Transmission Expansion Planning using Artificial Intelligent Tools

by

Tawfiq Taher Al-Saba

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

November, 1999

Transmission Expansion Planning Using Artificial Intelligent Tools

BY

Tawfiq Taher Al-Saba

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the

Requirements for the Degree of

MASTER OF SCIENCE

In

Electrical Engineering

November 1999

UMI Number: 1397410

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

بسم الله الرحمن الرحيم

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Tawfiq Taher Al-Saba**, under the direction of his Thesis Advisor and approved by his Thesis Committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE in ELECTRICAL ENGINEERING.**

Thesis Committee

Dr. Ibrahim M. El-Amin (Chairman)

Dr. Maamar Bettayeb (Member)

H. Duwaish

Dr. Hussain Al Duwaish (Member)

Z. Hamouz

Dr. Zakariya Al-Hamouz (Member)

Dr. Chokri Belhadj (Member)

Dr. Samir A. Al-Baiyat
Department Chairman

Dr. Abdallah M. Al-Shehri
Dean of Graduate Studies
Date: 17-11-99

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

الإهداء

أهدي هذه الرسالة :

إلى سيد المخلوقات أجمعين من آدم إلى يوم الدين

إلى الممدوح في القرآن بالخلق العظيم . . . الصادق الأمين

إلى خاتم الأنبياء والمرسلين . . صاحب الحضرة القدسية وزعيم الأمة

المحمدية... *الرسول المصطفى محمد وآله وصحبه أجمعين*

إلى الكف الممتلئة حنانا ودفئا والقلب الواسع الرحيب . . . *والدتي*

إلى البحر الزاخر والمؤدب الحكيم الموقر . . *والدي*

إلى شريكة الحياة ورفيقة الدرب . . *زوجتي*

إلى زملاء الطفولة وشركاء العيش . . *إخوتي وأخواتي*

Acknowledgment

Acknowledgement is due to King Fahd University of Petroleum and Minerals for the support of this research.

I would like to express my appreciation to Dr. Ibrahim El Amin who served as my thesis advisor. I also wish to thank the members of my Thesis Committee Dr.Hussain Al-Duwaish, Dr. Maamar Bettayeb, Dr. Zakariya Al-Hamouz and Dr. Chokri Belhadj for their valuable advices thoughout the research.

Also, I would like to thank my Family and my friends for their support and encouragement to complete this research.

Table of Contents

List of Figures

List of Tables

Thesis Abstract

Name of Student : Tawfiq T. Al-Saba
Title of Study : Transmission Expansion Planning Using Artificial Intelligent Tools
Major Field : Electrical Engineering
Date : November, 1999

The Transmission Expansion Planning Problem was solved using the Artificial Intelligent (AI) methods. These methods combine three techniques, which are Artificial Neural Network (ANN), Genetic Algorithm (GA) and Tabu Search (TS). The main goal is to minimize the investment and the power loss costs in a way that satisfies the defined constraints as shown in the objective function.

These methods were tested and compared with the Linear and Quadratic Programming Model through the application of a small (six-bus) system. The performance was also checked through the IEEE – 25 Bus System. The results showed the AI methods are reliable for the solution of the problem. Moreover, the method was used to determine the next ten years expansion plans for the SCECO – East's 380KV network.

Master of Science Degree
King Fahd University of Petroleum and Minerals
November 1999

خلاصة الرسالة

اسم الطالب / توفيق طاهر علي السبع

عنوان الرسالة / تخطيط شبكات الضغط العالي باستخدام أدوات الذكاء الصناعي

التخصص / هندسة كهربائية

تاريخ الشهادة / نوفمبر ١٩٩٩

تتلخص هذه الرسالة في إيجاد طريقة لحل مشكلة تخطيط شبكات الضغط العالي باستخدام طريقة حديثة سميت بـ(أدوات الذكاء الصناعي) ، التقنية التي دخلت في تصميم هذه الطريقة هي الشبكة الدماغية الصناعية والنظرية الجينية وطريقة البحث الرياضي المعروف بالبحث في المنطقة المحرمة ، الهدف المرجو من ذلك هو تخفيف تكلفة الاستثمار الاجمالية وتقليل تكلفة المفقود من الطاقة خلال عملية النقل مع مراعاة بعض الموانع التقنية والاقتصادية التي يجب أخذها بعين الاعتبار .

تم تجربة هذه الطريقة من خلال استخدام شبكة صغيرة مكونة من ست محطات وكذلك تمت مقارنتها مع طريقة البرمجة الخطية وكذلك البرمجة الغير خطية ذات الدرجة الثانية ، وللتأكيد من موثوقية هذه الطريقة في حل مشكلات الشبكات ذات المحطات الكثيرة ، فإنه تم تطبيقها على شبكة الجمعية العالمية للمهندسين الكهربائيين والإلكترونيين ذات الضغط العالي (٢٣٠ و ١٣٨) كيلو فولت المكونة من (٢٥) محطة ، وبما أن النتائج أثبتت بأنها طريقة ناجحة ذات إطار تنافسي مع الطرق الأخرى في حل هذا النوع من المشاكل فإنها طبقت على شبكة الضغط العالي (٣٨٠)كيلو فولت لكهرباء الشرقية وقدمت عدة تخطيطات مقترحة للعشر سنوات المقبلة.

درجة الماجستير في العلوم

جامعة الملك فهد للبترول والمعادن

الظهران – المملكة العربية السعودية

CHAPTER - 1

INTRODUCTION

Electrical power systems play a major role in modern society by providing the means for generation, transmission and distribution of electrical energy [1]. The key objective of an electrical utility is to minimize capital and operating costs required to provide an adequate level of reliability, with due consideration to environmental and other related issues.

A main component of the power network is the transmission system. It plays an important role in the operation and planning of electric power systems [2-3]. It includes all lands (right of ways), transmission lines and switching equipment that are used to connect the power at the generation points to the receiving points at the user load center. Also, it is important because of three main reasons. The first is its ability to transmit power over a very long distance between the generation plants to load centers. Second, it is needed to evacuate bulk quantities of power to load centers from outlying power stations. Finally, it is necessary for the transfer of energy from one system to another in case of emergency or in response to diversity in system demand.

This thesis deals with the problem of transmission system planning. It is an attempt to use of new tools for the solution of the problem. This chapter presents and defines the transmission expansion problem. It will also persent the thesis motivation and outline.

1.1 *Transmission System Planning.*

The purpose of transmission system planning is to determine the timing and type of new transmission facilities [4-6]. The facilities are required in order to provide adequate transmission capacity to cope with future generating additions and power flow requirements. The starting point of the planning procedure is to develop load forecasts in terms of annual peak demand for the entire power system. The development of generation plans to meet the demand forecasts is the next step in system planning. The total demand is then divided into regions, load centers and substations for distribution purposes. A transmission system is envisaged to evacuate the generated power into the load centers. The transmission plans may require the introduction of higher voltage levels, the installation of new transmission elements and new substations. The transmission plans must be designed to meet the system load at the lowest possible cost.

The planner must also identify the potential problems, in terms of unacceptable voltage conditions, overloading of facilities or the decrease of reliability [3-6]. Therefore, he has to develop alternative plans or scenarios that will prevent unforeseen problems and meet the long-term objectives of system reliability and economy. Moreover, the planner should also study system behavior under fault or short circuit conditions. A stability study has to be conducted to test system performance.

The setting of minimum planning criteria or system performance standards is essential, both for deciding when the existing system will be inadequate and for ensuring that proposed development schemes will be satisfactory [4-6]. So, a transmission system planner must consider the utility's standards during the design development. The standards will include:

- The thermal rating limit of any system element should not be exceeded under specified normal and emergency operating conditions.
- Voltage level should remain between specified operating limits at light load and at maximum load.
- Accepted limits of reactive source capability should not be exceeded, e.g. generator under-excited var limits, under specified operating conditions, with defined safety margins.
- Standard levels of short-circuit current should not be exceeded at defined operating conditions.
- A system should be able to withstand, without instability ensuing, a specified severity of system fault.
- The damping characteristics of the system should meet specified criteria.
- Specified limits to the magnitude of voltage fluctuations due to e.g. arc furnaces, petrochemical loads, etc.
- Frequency deviations should be within specified limits.
- Phase unbalance at various voltage levels should not exceed specified limits.

- Harmonic voltages should be appropriately limited.

Transmission system expansion is an attempt to determine and arrange the timing and the type of new facilities so that the total investment costs are minimized and so that the above standards criterion are met

1.2 *Thesis Motivation*

Transmission system planners have used many methods to solve the expansion problem. The techniques range from human and engineering judgements to powerful mathematical programming methods. The engineering judgements depend on human expertise and knowledge of the system. Mathematical programming methods provide accurate solutions within long computational time. This research is to provide new powerful methods that obtain relatively accurate solutions within a short time.

Recently, many Artificial Intelligent (AI) algorithms have been used to solve the power system problems. The AI methods include Genetic Algorithm (GA), Tabu Search (TS), Artificial Neural Network (ANN), ..., etc. These methods proved to be reliable and they provide optimum solutions within a short computational time.

This thesis reports on the work undertaken to address the Transmission Expansion Problem (TEP) using the AI tools. The objective of the work described in this thesis is to apply methods such GA, TS and ANN for the solution of the transmission expansion problems and to investigate their performance against Linear Programming and quadratic Programming methods.

1.3 *Thesis Outline*

The thesis is organized in six chapters. Chapter one is the introduction. Chapter two includes a literature survey of the work undertaken by researchers and investigators. The mathematical formulation of the expansion problem is given in chapter three. Chapter four introduces the basics of the algorithms that are going to be used while chapter five presents the results of the application of the Artificial Intelligent methods to several power systems. Finally, chapter six contains the thesis conclusions and recommendations. Also, a list of references and a number of relevant appendices are included at the end of the thesis.

CHAPTER-2

LITERATURE REVIEW

Transmission system planners, worldwide, tend to use many methods to address the expansion problem. The planners utilize automatic expansion models to determine the optimum system. In a mathematical sense, an expansion model will minimize an objective function subject to a number of constraints. The objective is to minimize the cost of the transmission system. The constraints include system-operating indicators.

The automatic expansion formulations can be classified into the following [3]:

1. Single stage "Static" mathematical optimization model.
2. Time-phased "Dynamics" mathematical optimization model.
3. Heuristic Models.
4. Artificial Intelligent Algorithms.

This chapter will highlight the contribution that has been done in the fields of transmission expansion planning.

2.1 *Static Mathematical Optimization Model*

Single-stage static optimization models are used to determine the optimum network expansion from one stage to the next. They generally do not consider the timing of the expansion. They provide an optimum solution for year-by-year expansion. However, they may not give an optimum solution for the overall expansion pattern over an entire time horizon [3]. The mathematical programming techniques used in this model include:

1. Linear Programming.
2. Integer Programming.
3. Quadratic Programming
4. Decomposition Approach.
5. Gradient Search Method.

2.1.1 Linear Programming Model:

Linear Programming (LP) can be used to minimize a given linear objective function, in which the variables are subject to linear constraints. The objective function takes the following linear form [3]:

$$Z = \sum_{i=1}^{n} c_i X_i \qquad (2.1)$$

where Z is the value to be optimized, the $X_i^{'S}$ represent the unknown quantities and the $C_i^{'S}$ are the coefficients most associated with unknowns. The constraints or restrictions are limitations on the values. The constraints are

$$\sum_{j=1}^{n} \mathbf{a}_{ji} \ \mathbf{X}_i \qquad (2.2)$$

j=1,2,...,m

Where

X_i : represents n-unknown quantities

a_{ji} & b_i : are the coefficients most associated with unknowns

n : represents the number of unknowns

m: represents the number of constraints equations.

The constraints define a region of solution feasibility in *n*-dimensional space. The optimum solution is the point within this space whose X_i values minimize or maximize the objective function **Z**.

LP was used to solve the Transmission Expansion Problem (TEP) [7-10]. The objective was to discount the capital and operating costs associated with the system expansion over the planning horizon. The constraints associated with this model are essentially the physical and economical constraints.

In 1988, LP was used based on a maximum principle for an expansion planning problem by Ken J. Kim and others [8]. They proposed a new cost model and an analytical expression for the investment managerial cost. Their basic objective was to find the type and location of the network additions at minimum cost subject to reliability, and environmental, legal, and political constraints. They used the model to minimize the operation cost under the given constraints where the operating cost could be made of fuel and failure costs. Moreover, this problem's solution has not only considered normal operation, but also included contingencies due to changes in the system. A hypothetical five-bus transmission system was considered for testing the proposed algorithm. The results of the optimal investment algorithm for a 5-years planning problem were presented.

Another contribution that uses LP was done by R. Villarana and et al in 1985 [9]. They formulated the problem in a mathematical sense. They modeled the network in such away as to use a DC-power flow. They considered the following constraints during the analysis:

1. The power balance at each bus, the power flow into the bus equaling to the power flow out;
2. The line power flow limit, the capacity of the lines not being exceeded.
3. The system power balance, the generated power being able to handle the demand and the power losses.

Their main objective was to minimize the capital cost. To illustrate their method, a six-bus system was used. The final equipment cost in the test was 200 monetary units, corresponding to 200 miles of the added transmission lines.

2.1.2 *Integer Programming Model*

Integer programming (IP) refers to the class of LP problems in which some or all of the decision variables are restricted to be integers. For example, in order to formulate the LP program given in an equation (2.1) and (2.2) as an integer program, a binary variable is introduced for each line to denote whether it is selected or not:

$X_i = 1$ if line is selected

$X_i = 0$ if line is not selected

$$\textbf{Minimize} \quad Z = \sum_{i=1}^{n} C_i X_i \qquad (2.3)$$

$$\textbf{Subject to} \quad \sum_{i=1}^{n} a_{ij} X_i = b_j, \qquad X_i = 0,1 \qquad (2.4)$$

Where $\quad j = 1,2 \ldots\ldots, m, \quad i = 1, 2 \ldots . n$

Generally, IP is more suitable for the TEP than LP. This is because it takes into account the discrete nature of the problem, i.e. a line component is either added or not [11].

As an example of using an IP model, a static network synthesis method was applied to a transmission expansion planning problem. It considered the minimization of network expansion costs subject to constraints that assured the transmission capacity for load supply. The variables described the network configuration with its transmission capacity and the associated power flows. A proposed synthesis was formulated as an optimization model which was solved by an efficient algorithm of network flow which fixed the charge [12]. The problem was formulated considering both economical objectives and power flow constraints. Power flow accuracy was obtained taking implicitly into account the DC-load flow on the optimization model. The proposed model was solved by an implicit enumeration search procedure. The methodology of the proposed method was applied to the example of a TEP proposed by Garver to provide an alternative with low cost and good electrical performance [12]. Using the exact DC-flow for the resulting network configuration, the final investment cost in the application was 200 monetary units, corresponding to 200 miles of the added transmission lines.

Also, as an another example of the IP model, a formulation for optimum HVDC-transmission expansion planning was applied to the same six-bus system [13]. The basic objective of the HVDC-formulation was to obtain the optimal configuration that minimized the annual amortized cost function including the investment cost for equipment additions and the operation cost in terms of the real power transmission losses. This formulation also used the implicit enumeration formulation that allows for an exact treatment of the discrete nature of equipment additions and used mixed integer linear programming. The DC-network equations were exact, and sensitivity coefficients were not required in this formulation. The formulation was quite appropriate and applicable for HVDC-transmission expansion planning.

2.1.3 Quadratic Programming Model:

Quadratic Programming (QP) can also be used to minimize or to maximize a given quadratic objective function. It is a more modified form of the linear programming and its variables are subject to given constraints. The objective function takes the following form [14]:

$$Z = \sum_{i=1}^{n} C_i X_i + \sum_{i=1}^{n} D_i X_i^2 \qquad (2.5)$$

Where Z is the value to be optimized, the X_i's represent n-unknown quantities and the C_i's and D_i's are the coefficients associated with unknowns. The constraints or restrictions are limitations on the values. The constraints assume for (2.5)

$$\sum a_{ji} X_i = b_i \qquad (2.6)$$

$$j=1,2,\ldots,m \qquad i=1,2,\ldots,n$$

The main feature of this formulation is that it includes more factors in the objective function [14]. In 1989, Al-Hamouz applied this method to a six-bus system and to the Jordan power system. The objective was to obtain the exact cost of power losses and capital investment in the new facilities. Through the use of DC-load flow, the results were comparable to the LP and IP approaches. It also gave lower cost than the cost proposed by the Jordan utility.

2.1.4 Decomposition Approach

The Decomposition approach is another approach that can be used to solve the optimization problem [15-17]. The main concept of this approach is to divide the problem or the objective function into sub-problems under the limits of defined constraints. The minimization decision process of the problem is based on the solutions of sub-problems where these solutions are obtained by dependent stages.

This approach was applied to transmission system planning problems. It divided the problem into two parts; the first one is to determine the optimal investment in new system capacity, while the other one is to determine the system operating cost and supply reliability associated with construction of this new capacity. These techniques achieve the global optimization of the investment and expected operation costs through the iterative solution of investment and operation sub-problems. In other words, the decomposition techniques allow the investment and operation sub-problems to be modeled separately using different solution algorithms. This approach has many advantages in term of flexibility, modularity and consistency. For example, the same production costing module used for operation planning studies could be used as part of an automatic generation expansion program. Also, an optimal power flow program could be used as a sub-model of an automatic transmission expansion program or for optimal, reactive capacity planning. The investment sub-problems can benefit from specialized solution techniques if solved separately from the operation sub-problems. In 1985, M.V. Pereira and others applied this approach to the Southern Brazil System [15]. They divided their problem into two sub-problems which were investment and operation sub-problems. They solved these sub-problems using DC-power flow and they used two options. The first one was to model the problem as LP while the other one used IP. Although they got different results using the two methods, this approach proved that it could be used for any transmission system problem and give a feasible solution with less cost.

A decomposition based method was applied by Vikto A. Levi and Milan S. Calovic in 1991 for the Yugoslavia power system [16]. Their results were acceptable in terms of economical and technical constraints. The sub-problem of initial load flows was solved by using the model of minimum load curtailment (MLC) which minimized the total non-supplied load in the system by respecting generation and branch loading limits. The mathematical representation was modeled using LP and the DC-power flow was considered. The sub-problem of superimposed load flows used the proposed model of marginal network (MN). Under certain circumstances, the operation model was solved by using the Monte Carlo simulation. Particular attention was paid to the operation cost that originates from the transmission network constraints. The model included additional reliability constraints and applied control strategies.

R. Romero and A. Monticelli developed their proposal which was a hierarchical decomposition approach to solve the transmission expansion planning problems [17]. The basic concept of this approach was similar to previous approaches except that they implemented their approach using three different levels of network modeling, which were the DC-power flow model, the transportation model and the hybrid model. Their proposal was illustrated through an expansion problem and the resulting sub-problems (operation sub-problem, and investment sub-problem) were solved alternately using the three models until convergence was attained. Moreover, the investment sub-problem was solved by using linear programming while the operation sub-problem was solved via a version of the algorithm.

2.1.5 Gradient Search Method:

The gradient search method is a nonlinear mathematical programming technique. It was applied to automated transmission system planning [18]. The objective function of the given transmission network is a performance index such as fault currents or system losses. The method starts with a DC-load flow solution for the initial transmission network and future load and generation forecasts. The system performance index is calculated and the necessary circuit modifications are made employing the partial derivatives of the performance index with respect to circuit admittance. Again, a DC-load flow solution is obtained, and the procedure is repeated as many times as necessary until a network state is achieved for which no further decrease in the performance index can be obtained.

As an example of this method, an oscillatory stability consideration was applied to a transmission expansion planning problem by Yuan-Yin Hsu and others in 1989 [18]. The index performance of this approach was frequency oscillations and they applied it to the Taiwan Power System. The analysis of oscillatory stability was performed using eigenvalues values and eingenvactor. They were employed as a basis for the justification of various expansion plans. The results concluded that the less the power flows over the trunk lines, the better the damping for the low frequency electromechanical mode. Other factors, like maximum power transfer limits, fault current, transient stability, and the economical consideration, were also examined.

2.2 *Time Phase Mathematical Optimization model*

Unlike static transmission network expansion models, time-phased or dynamic optimization models take into account the timing of new installations through a given time horizon [19]. Dynamic optimization models can include inflation, interest rates as well as yearly operating costs in comparison of various network expansion plans.

In 1993, El-Metwally and Harb developed a method that can be used in a dynamic model [19]. The method was based on using the admittance approach and quadratic programming. The problem was solved under the investment of cost, losses, load flow and security constraints including interest and inflation rates. The six-bus system had been used as an application of the proposed algorithm for static and dynamic cases. As a conclusion, the final network designed during the static approach was different from that designed using the dynamic approach. Also, the total cost resulting from the design using the dynamic approach was less than that using the static approach because the dynamic design was made so as to follow the load growth.

2.3 *Heuristics Method :*

In the heuristic approach , the best circuit addition or exchange is given by the computer program automatically at each stage of the network design process. The planner can accept it or modify it as desired. The model procedure can be chosen from among the models. The first one is the DC-load flow model. which satisfies the unit active power flow in each line under the limitation of the line capacity. The transportation model is the second. It satisfies the first Kirchoff law, which takes care of the distribution of the power flow in the network at each bus. The last model is the hybrid model which combines the characteristics of the DC-load flow and the transportation models [20].

During the analysis of the network expansion, the model is formulated and then new equipment additions are stated and calculated. This procedure is repeated until a configuration compatible with load and generation levels is achieved. Generally, this approach considers that the economical criteria can be satisfied at each step of the process, then the addition that presents the lowest cost/benefit ratio can be selected. Moreover, the advantage of heuristic models are interactive planning, simplicity and logic. The planner can observe the expansion process and direct it as desired.

Chopin is an example of the Heuristic Model [20]. In 1993, Gerardo Latorre and Ignacio Perez formulated the system expansion problem to minimize the global annual cost. The solution method utilized the natural decomposition between the investment and operation sub-models & the network was presented by a DC-load flow model.

The solution method started from a user-provided initial expansion plan in which the algorithm improved an initial plan by performing local searches. Each of the plans that were provided by the heuristic search was evaluated by the corresponding sub-problem modules in order to determine its investment, production and reliability costs.

Generally, The heuristic search procedure in CHOPIN was guaranteed to avoid repetitions in the evaluation of the plans and without truncation of the solution space. It yielded exhaustive enumeration of the solutions. Unfortunately, truncation was unavoidable in network expansion problems of realistic size although there was no evidence of that truncated enumeration procedure in CHOPIN.

2.4 *Artificial Intelligent Tools*

The TEP can be solved using Artificial Intelligent tools (AI). The main feature of these search tools is that they can converge to the optimal solution starting from random solution states. The AI methods can also provide the optimal solution with less computation time and with the same or better accuracy relative to other optimization tools. Moreover, it can be modeled in a very simple manner to solve a very large system scale. These tools include Simulated Annealing, Genetic Algorithm, Artificial Neural Network, Tabu Search, and Fuzzy Logic.

2.4.1 *Simulating Annealing*

The basic concept of the Simulating Annealing (SA) approach is to have thermal equilibrium during the analysis stages [21-22]. The SA algorithm can be stated as follows. The first step is to initiate the required variable, like the temperature, and assume the current solution as an optimum one. The next step is to evaluate the cost function for the new variable and to update incumbent solution if it is less than the old one. The acceptability of the solution is checked through the comparison between random numbers, the exponential function of temperature and cost variation. Moreover, a thermal equilibrium test and temperature reduction must be conducted along with updating the new solution if the solution is to be excepted. A good characteristic of the SA algorithm is that in addition to the optimal solution it provides a number of interesting local solutions found during the simulation process. These near-optimal solutions can be further analyzed by the planner to take into account factors other than expansion costs. SA was applied to a TEP [21-22]. The model was formulated as a mixture of integer and nonlinear programming problems. The network was represented by a D.C. power flow model.

In 1996, R. Romero and others applied the SA method to a six bus system [21]. The results obtained with this approach had been validated through comparison with the results given by other optimization approaches. In this case, the optimal solution was to add seven lines distributed among the network links with a cost of 200 monetary units.

In 1997, R. Romero and others also applied SA using multi-processors in the computer machine [22]. This method is called Parallel Simulated Annealing. The main purpose of this approach is to speed up reaching thermal equilibrium. This means having the optimum solution with less computational time. They applied this approach to the same six-bus example and they got the same result as in a sequential SA algorithm. Their conclusion was that if more processors were added to the machine, less time would be required to complete the search for the optimal solution.

2.4.2 *Genetic Algorithm*

Genetic Algorithm (GA) has the ability to find near-optimal solutions; however, the merit of GA will be weakened when it considers inequality constraints. Generally, GA is an excellent tool to perform global search. The concept of GA is that the states of solution are allowed to exist in the population. Also, its operation is carried out until the population converges to a global solution. The following are used as genetic operators.

Selection: uses of roulette selection and elite preservation are combined.

Crossover: one or more points crossover.

Mutation: selected one bits are changed to opposite state.

GA was used in 1996 in dynamic transmission planning methodology [23]. The purpose of using GA was to determine an economically adopted elective transmission system in a dergregulated open access environment. GA tried to minimize the objective function by including variable costs of generation investment and losses. The Chilean electric system was chosen to apply the proposed approach. It had been studied in a ten year horizon. This system was characterized by its radial longitudinal structure, including most of its installed generation capacity. The solutions were under the consideration of active power flows and security constraints. Results were obtained from initial plans that corresponded to one of the initial populations. Moreover, the new members of the population were created by the random generated changes and by the ones determined from discrete sensitivities.

Also, In 1998, Gallego and etal applied the GA to North-Northeastern and Southern systems in Brazil [24]. Since these systems have unknown optimal solutions, the GA provides less cost comparison than other models.

2.4.3 *Artificial Neural Network (ANN) and Genetic Algorithm (GA)*

Through the use of Hop-Field type, the fundamental purpose of the ANN is to get 0-1 output that minimizes the scalar cost function of the network error. Then, this output will be applied to the problem to be minimized or maximized with the consideration of the constraints. This means that ANN allows fast preparation of solution states but it has been pointed out that it is difficult for ANN to find an optimal solution.

Both ANN and GA were used to prepare a plural number of solutions for solving optimization problems [25]. GA and ANN were used, in 1995, in the field of transmission expansion planning. That was done by using neuron-computing hybridized with GA. In this method, the authors hybridized ANN and GA with a different formulation that was a mixture of both to get the optimal solution. The six-bus system was used to show the effectiveness of this approach. The approach could find many good solutions in a reasonable time.

2.4.4 Tabu Search (TS)

The main feature of Tabu search is that it uses memory structure to direct the search process, with use of neighborhood strategy. In 1998, R. Romero and et al proposed a new method [26]. This method hybrids the basic concept of TS with SA and GA. However, the application to North-Northeastern and Southern systems in Brazil was discounted compared to GA and SA results during each of their application individually.

2.4.5 Fuzzy Logic

To best of the author's knowledge, no contribution has been done in the field of TEP using Fuzzy Logic.

CHAPTER - 3

PROBLEM FORMULATION

Power system transmission planning addresses the problem of determining the optimal number of lines that should be added to an existing network to supply the forecasted load as economically as possible, subject to operating constraints. The objective is a minimum cost expansion plan given the base network configuration, the generation facilities, and the forecasted demands for a target year.

Transmission planning always involves consideration of the restrictions of the expansion scenario that can affect decisions taken when a new transmission line is built. As a result, the expansion planning tools that have been used in a utility have to be adaptive to handle the changes to the new environment conditions. This means that these planning tools have to fit the new demands without losing their power to solve the day-to-day problems.

This chapter will discuss, in detail, the mathematical representation of the Transmission Expansion Problem (TEP) considering how to fit the required in-terms of operation and the investment costs. It will also define the constraints that should be considered during the planning stages.

3.1 *Problem Statement*

The TEP has to be addressed on several points. Generally, these points deal with the operation of the new transmission facilities that will be installed and the economics of the capital cost that results from the cost investment aspects. The objective is to minimize the capital and operating costs associated with the physical and economical constraints. These are important when attempting to expand a utility system at a minimum cost and yet to meet all economic and demand restrictions that are placed upon the system. The TEP can be formulated in the following terms:

$$\min \quad v = \sum_{i=1}^{NB} \sum_{j=1}^{NB} c_{ij} n_{ij} + K \sum_{i=1}^{NL} I_i^2 R_i \qquad \textbf{(3-1)}$$

Where

c_{ij} : is the cost of the additional circuits in branch i-j

n_{ij} : is the number of circuits added to the branch i-j

NB : is the total number of buses in the system

K : is the loss coefficient, K = 8760 x NYE x $C_{kwh.}$

NYE : is the estimated life time of the expansion network (years)

C_{kWh} : is the cost of one kWh (SR /kWh).

R_i : is the resistance of the ith line.

I_i: is the flow on the ith line.

NL : is the number of the existing lines.

Equation (3.1) is a typical hard combinatorial problem. It is prone to combinatorial explosion as the number of decision variable increases. An extra complication relates to the fact that there are cases in which the planning does not simply mean the reinforcement of an existing network. To guarantee the network connectivity, the combinatorial burden is even heavier than it would be in simpler, reinforcement only types of problems.

The loss coefficient (K) depends on the number of years of operation and the cost of kWh, which means that as the number of years of operation an/or the cost of kWh increases, the loss coefficient increases.

The DC-load flow is used in problem formulation where the current I (as in equation (3.1)) is approximated to be equal to the power flow, and voltage is assumed to be 1 in all buses. The DC-load flow model is given in the following section.

3.2 DC- Load Flow

To solve the TEP, the DC-load flow model is to be used. Although AC-power flow gives accurate operation cost, the DC-power flow is viewed as sufficient for planning purposes because of its simplicity and much less computation time. Consider a general representation of the AC-power flow given by the following equations:

$$P_i = V_i \sum_{k=1}^{NB} V_k [G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k)] \quad \textbf{(3-2)}$$

$$Q_i = V_i \sum_{k=1}^{NB} V_k [G_{ik} \sin(\theta_i - \theta_k) + B_{ik} \cos(\theta_i - \theta_k)] \quad \textbf{(3-3)}$$

Where

P_i: real power of bus i,

Q_i : reactive power of bus i,

V_i : voltage magnitude of bus i,

θ_i : voltage phase angle of bus i,

G_{ik} : real part of element (i, k) of bus admittance matrix,

B_{ik}: imaginary part of element (i,k) of bus admittance matrix,

N_B: total number of buses.

For DC- load flow model, we normally make the following assumptions:

(i) $Vi \approx 1$ for all i busbars .

(ii) $\sin(\theta_i - \theta_k) \approx \theta_i - \theta_k$, $\cos(\theta_i - \theta_k) \approx 1$,

(iii) $G_{ik} \approx 0$.

Then the AC-power flow equation (3-2) is simplified to yield the following DC-power flow equation:

$$P_i = \sum_{k=1}^{NB} B_{ik} (\theta_i - \theta_k) \qquad i = 1, \ldots\ldots, NB \quad \textbf{(3-4)}$$

Where B_{ik} is the line susceptance between bus i and k.

3.3 Transmission Expansion Constraints

Any power network consists of at least one generator, one load, and one transmission line. Therefore, there are three requirements that a network has to satisfy:

1- Generator(s) must meet the demand, i.e., the total generator output in the network must be greater than or equal to the load.

2- Thermal limits of the new and existing lines can not be exceeded when running a power flow for the new network system.

3- There must be one or more transmission lines, either existing or possible candidates, to connect all the buses in the network.

The restrictions have to be modeled in a mathematical representation to ensure that the mathematical solutions are in line with the planning requirements. This section will explain the transmission problem constraints that have to be satisfied during the planning. These are as follows:

1. *Limits in branch power flow*:

$$|\mathbf{P}_{il}| \leq \mathbf{P}_{il}^{max} \qquad \textbf{(3-5)}$$

Where P_{il}^{max} is the maximum branch power flow between the buses i and l.

In a DC-load flow model, each element of the branch power flow in constraint (3-5) can be described as follows:

$$P_{il} = (Z_{il} / x_{il})(\theta_i - \theta_l) \qquad \textbf{(3-6)}$$

Where

Z_{il} : is the variable representing the total number of parallel links between the buses of i and l

x_{il} : is the reactance of link of branch i to l

θ_i : is the voltage angles of the terminal buses of branch i.

Then, to generalize it in matrix form, constraint (3-5) becomes:

$$B \,.\, \theta = P \qquad \textbf{(3-7)}$$

Where B is the susceptance matrix whose elements are $B_{il} = -\,(1/X_{il})$ for the off-diagonal terms, and $B_{ii} = \sum_{l \in \Omega_i} B_{il}$ for the diagonal terms,

X_{il} : is the total reactance of branch (i,l) ,

$l \in \Omega_i$: are the branches connected to bus i,

θ is the vector of nodal voltage angles.

2. *Bus Voltage Angle.* Since The DC-load flow model is going to be used, the voltage bus magnitude will not have any effect in the analysis. The constraint of the voltage angle will be considered. This constraint can be stated as the calculated phase angel at bus i θ_i^{cal} which should be less than the maximum phase angle θ_i^{max}.

$$|\theta_i^{max}| \geq |\theta_i^{cal}| \qquad \textbf{(3-8)}$$

3. Right-of-way:

It is important to know the line location and the capacity of the new required lines. This is because the planners have to meet the community standards of visual impact on the environment along with the economic considerations. So, the acquisition of right-of-way has to be considered by the planner and should be included as a constraint of the problem. Also, the number of the transmission lines in the route has to be limited and phase spacing & ground clearance should be chosen to allow an acceptable safety condition for the surrounding environment. Finally, the lead time from planning to the commissioning of the line has to be lengthened to allow for the preparation and publication of an impact statement and sometimes for a public inquiry. Mathematically, This constraint defines the line location and the maximum number of the lines that can be installed in the specified location and this can be represented by :

$$\mathbf{0 \le n_{ij} \le n_{ij}^{max}} \qquad \textbf{(3-9)}$$

Where

n_{ij}: is the total number of circuits added in branch i-j, $n_{ij} = \frac{x_{ij}}{\gamma_{ij}}$.

x_{ij} : is the total reactance added in branch i-j.

γ_{ij} : is the initial line reactance in the line i-j.

4. *Power nodal balance at each bus*

This constraint checks that there is no need for artificial generation to make the balance between the load, the line losses and the generation units. Mathematically, this is represented by:

$$\mathbf{g} = \mathbf{d} + \mathbf{B}.\theta + \mathbf{r} \qquad \textbf{(3-10)}$$

where

g: generation vector in the existing power plants.

d :load demand vector in all network nodes.

B: susceptance matrix whose elements are the imaginary parts of the nodal admittance of existing ones ($B_{ij}^{existing}$) and the added lines to the existing network (B_{ij}^{added}).

θ : bus voltage angle phase vector.

r : extra generation needed in case of high transmission losses or an unbalanced power system.

5. *Reliability of the added line*

The transmission planning problem should not only consider the normal operation but also should include contingencies due to changes in the system, e.g., generator outage, line outage, load uncertainties, etc. The generator outages have already been treated in terms of the expected power output in the generation planning problem. Similarly, the load uncertainties have also been treated in terms of the expected or mean load curves. Therefore, it only remains to quantify line outage in order to obtain the expected value of operation cost.

In general, most of the line outages are single line outages, and it is very rare to have multiple line outages. Therefore, it is sufficient to consider only single and double line outages without loss of generality.

3.4 ***Summary of the Problem Formulation.***

The general formulation of the expansion problem (TEP) can be expressed as:

$$\min \ v = \sum_{i=1}^{NB} \sum_{j=1}^{NB} c_{ij} n_{ij} + K \sum_{i=1}^{NL} I_i^2 R_i \qquad \textbf{(3.11)}$$

S.t.

$$\mathbf{r} + \mathbf{B}.\theta + \mathbf{d} = \mathbf{g} \qquad \textbf{(3.12)}$$

$$| \mathbf{P}_{ij}^{\ cal} | \ \leq \ |\mathbf{P}_{ij}^{\ max} | \qquad \textbf{(3.13)}$$

$$0 \leq \ \mathbf{n}_{ij} \ \leq \ \mathbf{n}_{ij}^{\ max} \qquad \textbf{(3.14)}$$

$$\Delta\theta^{cal} \leq | \Delta\theta^{cal} | \qquad \textbf{(3.15)}$$

$$\theta^{cal} \ \leq \ | \theta^{max} | \qquad \textbf{(3.16)}$$

Where

C_{ij} : cost of the additional circuits in branch i-j

n_{ij} : number of circuits added to the branch i-j

K : loss coefficient, K = 8760 x NYE x C_{kWh}

NYE : estimated life time of the expansion network (years)

$\mathbf{C_{kWh}}$: cost of one kWh (SR /kWh).

R_i: resistance of the ith line.

I_i: flow on the ith line or route.

B: susceptance matrix whose elements are the imaginary parts of the nodal admittance of existing ones ($B_{ij}^{\ existing}$) and the added lines to the existing network ($B_{ij}^{\ added}$).

θ : phase angle vector.

d : vector of loads in all network nodes.

g : vector of generation in the existing power plants.

Δθ: vector of phase angle difference across the existing branches.

r : extra generation needed in case of high transmission losses or an unbalanced power system

The TEP stated in equations 3.11 to 3.16 is to be solved by the Intelligent Tools that chapter four will present.

3.5 *Other Discussion Points.*

Although the rules of the TEP depend upon whether the problem of expansion is less expensive than the cost of other alternatives, there are some aspects that cannot be formulated through the mathematical models [27-31]. The economy of scale is one of these aspects [27-28]. It describes the situations where the average cost of producing an item decreases as the quantity produced increases. One typical example of economies of scale is a new projected transmission line triggered by a specific generation project. Since the incremental cost of the additional transmission capacity is low, it may make economic sense to invest additional capacity beyond the current need of the specific project.

The other point is the allocation cost and recovery that deals with how to allocate line investment costs among the participants in a fair way when there are different utilities connected through the wheeling transactions. Several allocation rules have been proposed in relation to this subject such as Postage Stamp [29]. This approach uses the rule of a flat amount per KW. It is based upon the assumption that a wheeling transaction is confined to flow along a specified continuous path through the wheeling company's transmission system. Also, Marangon Lima et al [30] proposed that each participant's share should be proportional to its impact on system transmission investment requirements. Under some simplifying assumptions, this scheme reduces to a modified MW-mile rule. Tsukamoto et al [31] have proposed a methodology to allocate the cost of transmission network facilities if there are wheeling transactions according to the transmission usage pattern.

CHAPTER- 4

PROPOSED ALGORITHMS AND METHODS OF SOLUTION

It is believed that the TEP in a power system could be characterized by two startegies. The first one is the economic decision while the other is the power operation condition. Based on these strategies, the best transmission design should be with a minimum cost and high electric performance.

This thesis aims to get the optimal design using a fast automatic decision-maker. An intelligent tool starts from a random state and it proceeds to allocate the calculated cost recursively until the stage of the negotiation point is reached. These intellegent tools, Genetic Algorithm (GA), Tabu search (TS) and Artificial Neural Network (ANN), are flexible to handle and easy to implement in this type of problem. Figure 4.1shows a conceptual block diagram of the proposed solution method.

GA, TS and ANN will be applied as solution tools for the TEP. This chapter presents an overview of GA, TS and ANN and how they are formulated to handle the TEP. The four hybridization methods of TS, GA and ANN will also be explained.

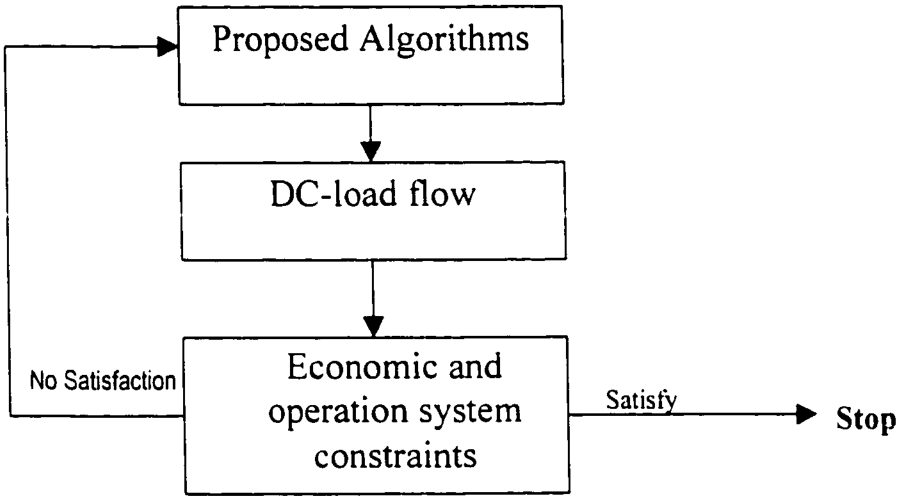


Figure 4.1 Block diagram of the analysis procedure

4.1 *Genetic Algorithm Approach*:

GA is a general-purpose search technique based on the principles inspired by the genetic and evaluation mechanisms observed in natural systems and populations of living beings [32-34]. Their basic principle is the solution (maintenance of a population) to a problem (genotypes) as encoded information individuals that evolve in time.

A genetic search starts with a randomly generated initial population within which each individual is evaluated by means of the fitness function. Individuals in these subsequent generations are duplicated or eliminated according to their fitness values. Further generations are created by applying GA operators. This eventually leads to the generation of high performing individuals.

Three major operators are usually used in a typical genetic algorithm. The first is the production operator (elitism) which makes one or more copies of any individual that prosess a high fitness value. Otherwise the individuals are eliminated from the solution pool.

The second operator is the recombination (also known as the "crossover") operator. This operator selects two individuals within the generation and crossover site and performs a swapping operation of the string bits to the right hand side of the cross over site of both individuals. Crossover operations synthesize bits of knowledge gained from both parents exhibiting better than average performance. Thus, the probability of a better performing offspring is greatly enhanced.

The third operator is the "mutation" operator. This operator acts as a background operator and is used to explore some of the invested points in the search space by randomly flipping a "bit" in a population of strings. Since frequent application of this operator would lead to a completely random search, a very low probability is usually assigned to its activation.

Formulation for the GA

The GA used in this research is the simple GA, and a plural number of identical individuals are allowed to exist in the population. The genetic operation is carried out until the population converges to an individual. Through its application to the TEP problem that was discussed in chapter three, offspring (chromosome) length repesents the number of the availible right-of-way in the nework while the offspring itself repesents the number of the newely added lines. Also, the cost of the new addition is represented in the fitness function. Moreover, the procedures of the GA are typically shown as follows:

1. Generate the initial population and calculate the fitness of respective individual.
2. Save the best individual.
3. End Algorithm if the old population converges as an individual.
4. Perform the repetition a-c until the number of offspring equals population size.
   a. Select two parents from old-population via roulette selection
      based on their fitness value.
   b. Perform the crossover to produce two offspring and mutation to all generation set.
   c. Calculate the fitness of the offspring and place it in the new population set.
5. Replace the old population by the new population.
6. Go to 2.

Figure 4.2 shows a flow diagram of the GA solution procedure.

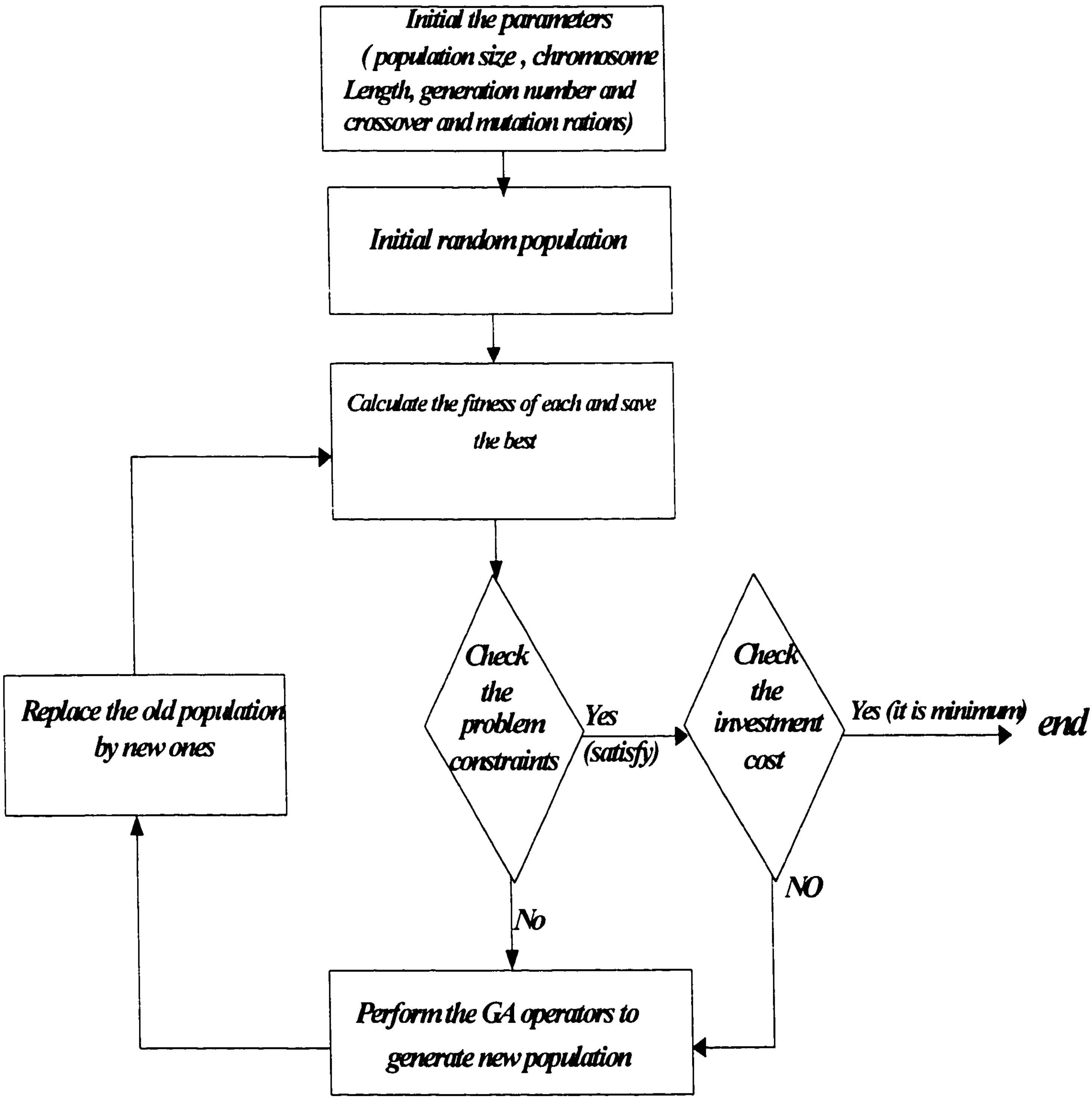


Figure 4.2 Flow Diagram of the GA Solution Procedure

4.2 *TABU Search :*

Tabu Search (TS) has its antecedents in methods designed to cross boundaries of feasibility or local optimality normally treated as barriers, and systematically to impose and release constraints to permit exploration of otherwise forbidden regions [35-36]. The philosophy of TS is to derive and exploit a collection of intelligent solution techniques. A fundamental element underlying the Tabu search is the use of flexible memory. From the standpoint of the TS, flexible memory embodies the dual processes of creating and exploiting structures for taking advantage of history. Beside this element, there is another important element, which is the way and the number of movements. The movement way can be done by random search or neighborhood search.

In neighborhood search, each solution $x^{now} \in X$ has an associated set of neighbors, $N(x) \subset X$, called the neighborhood of x^{now}. Each solution $x^{next} \in N(x)$ can be reached directly from x^{now} by a move operation . Normally in TS, neighborhoods are assumed symmetric; i.e. x^{next} is a neighbor of x^{now} if and only if x^{now} is a neighbor of x^{next}. However, the random movement can be implemented by any random move from the current solution x^{now} to the next solution x^{next}. In other words, there is no rule to perform the move operation and it can be done by any means. One more concern has to be mentioned which is the number of movements or the number of possible solutions when the movement is required from x^{now} to x^{next}. Actually, no specific rule is stated concerning this subject but it can be selected based on the feature of the optimization problem.

Formulation of TS

The TS search used in this work is a simple algorithm with a neighborhood search method. Through its application to the TEP problem that was discussed in the previous chapter, X is the solution vector that represents the number of new additions in each right-of-way. The cost of this new addition is represented in the objective function. The procedure of TS is as follows:

1. Initialize Tabu list size, movement set number, and minimum value as high values.

2. Initialize random solution set ($x^{now} \in X$).

3. Calculate the objective function.

4. Record the best solution as $X^{best} = X^{now}$ and best move = $C(X^{best})$.

5. If minimum value > best move, then minimum value = best move.

6. If best move $\notin$ Tabu List, then accept in Tabu List else don't accept.

7. Generate another random set of movements (x^{next}) such that X^{next} is neighbor to X^{now} ($X^{next} \in N(X^{now})$).

9. If the iteration limit is reached then end, or else go to 3.

Figure 4.3 shows a flow diagram of the TS solution procedures.

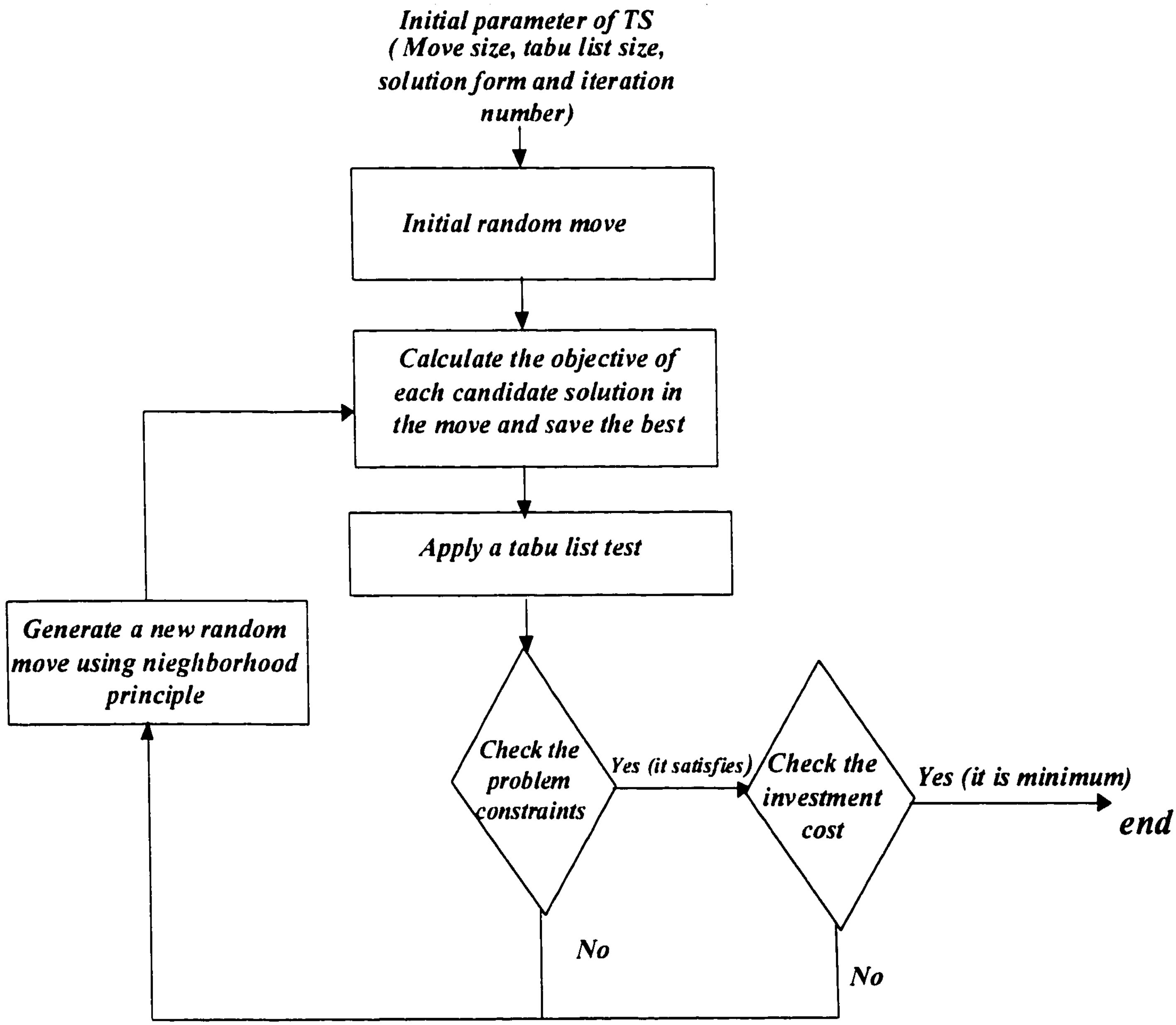


Figure 4.3 Flow Diagram of the TS Solution Procedure

4.3 *Hybridization Algorithm*

From the previous sections, it can be stated that the GA has the feature of combining the solution while the TS is a systematic exploration of memory function in search processes [37]. Also, both have the feature to become valuable enough for solving complex optimization problems. By taking advantage of their features and to improve the performance in term of obtaining the optimal solution, new mechanisms of solution search are going to be explained. The main feature of the modified ways is that they can create useful degrees of diversity and they can allow effective transition between feasible and infeasible solution regions. This section is going to explain two ways of hybridization between GA and TS. They will be as follows:

4.3.1 *Hybridization Of TS and GA (Model-1)*

In this method, TS will be applied with the same searching strategy as in section 4.2 but the best *n-solution set* will be stored. Then GA will be applied to start from the best *n-solution set as* an initial population. This is illustrated in Figure 4.4.

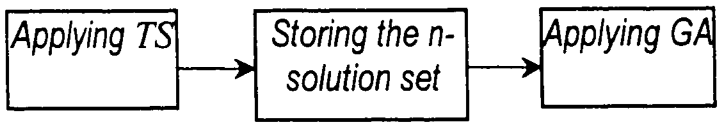


Figure 4.4 Hybridization algorithm (Model-1)

4.3.2 Hybridization (Model-2)

Since the neighborhood method was applied in the TS algorithm, the proposed way here will be to apply GA operators (Selection, Crossover and Mutation) in place of the neighborhood method. The proposed algorithm will be as follows. The first step starts from the random solution set then obtains the objective function and applies the stopping criteria. The second step is to apply the Tabu List acceptance test while the last is to apply the GA operators to generate a new solution set. Figure 4.5 illustrates this hybridization procedure.

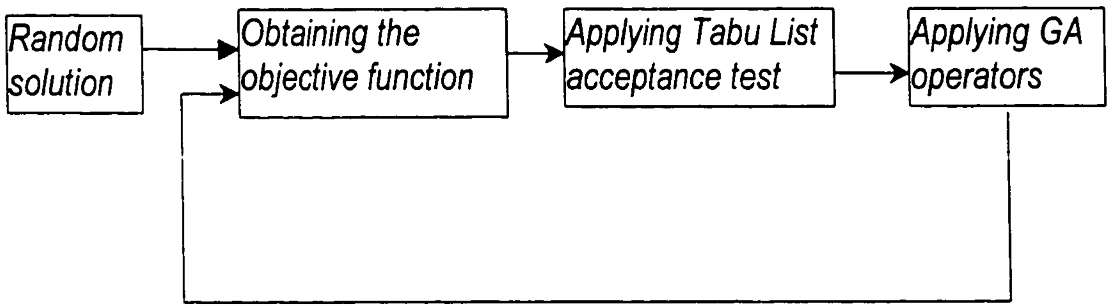


Figure 4.5 Hybridization (model-2)

4.4 Artificial Neural Network Model (ANN)

An ANN with a multi-layer perceptron model using a back-propagation algorithm is the proposed algorithm for TEP applications. It has some features that make it useful in the problem [38-41]. Some of these features are as follows:

- It has the ability to learn the relationship between the input and the output sets.
- Its has the ability to generalize its knowledge which means that it can recognize the patterns or sets which are similar to those with which they have had experience.

Moreover, back-propagation (BP) has also some features that make the operation of ANN more reliable [38-41]. The first feature is that the BP training is designed to minimize the means squared error between the desired output and the actual one across the whole training set. The other feature is that the BP is a supervised training technique.

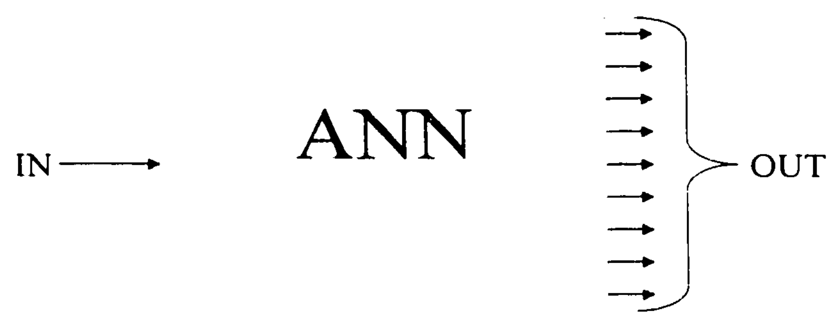
ANN Training Procedure:

A multi-layer perceptron with three layers is going to be chosen and a back-propagation algorithm will be used with off-line training for all input and output data. The proposed ANN has nine output neurons which represent the lines that should be added in each right of way while the input layer, which represents the fitness value (objective function value), is always one neuron (Figure 4.6). Moreover, the implementation procedure for the proposed ANN is as follows:

- Supplying the ANN with N training inputs.
- Obtaining the corresponding network outputs by a feel-forward structure.
- Upgrading the weights of the network according to the error between the desired and actual output.

These steps will be iterated until the network is well-trained which means that all the network weights can generate the Output/ Input sets. The final step will be to generate new states (or new scenarios) which were not in the training sets by applying a random value between zero and the minimum value of the training set.

Through the setting of the network parameters, it was decided to have three neurons in the hidden layer because it was found that an increase in the number of neurons does not produce any change in the output. Also, the network consumes a long time to finish the training. The normalization of Input / Output data was applied to prevent the simulated states from being driven too far into saturation. Once saturation is reached, changes in input values result in little or no change in output.



IN is a Random (0 - R) where R is the minimum value of the training set while the OUT is the output that represents the lines number that should be added to each right of way

Figure 4.6 Proposed ANN Architecture

Formulation of ANN

Through its application to solve the TEP problem in chapter three, output neurons repesent the solution state at each iteration while the state represents the number of the lines that should be added in each right-of-way in the nework. Also, the cost of these additions are represented in the objective function. The ANN procedure is as follows:

1. Initialize the state space size, iteration limit and the minimum value as high value.
2. Generate random initial states and calculate their objective function values to consider them as initial training set.
3. Perform the ANN training stage to obtain the network weights that correspond to the training set.
4. Generate a new state by applying a random value to the input layer (between zero and the minimum value of the training set) and calculate their objective function value, then record the best minimum value and store it.
5. If the minimum value > best value of each iteration, then minimum will be the best recorded value.
6. update the training set to have the states that containing the lowest objective values states
7. if the iteration reaches to the specified limit then end or else go to 3

Figure 4.7 shows the flow chart of the ANN solution procedure.

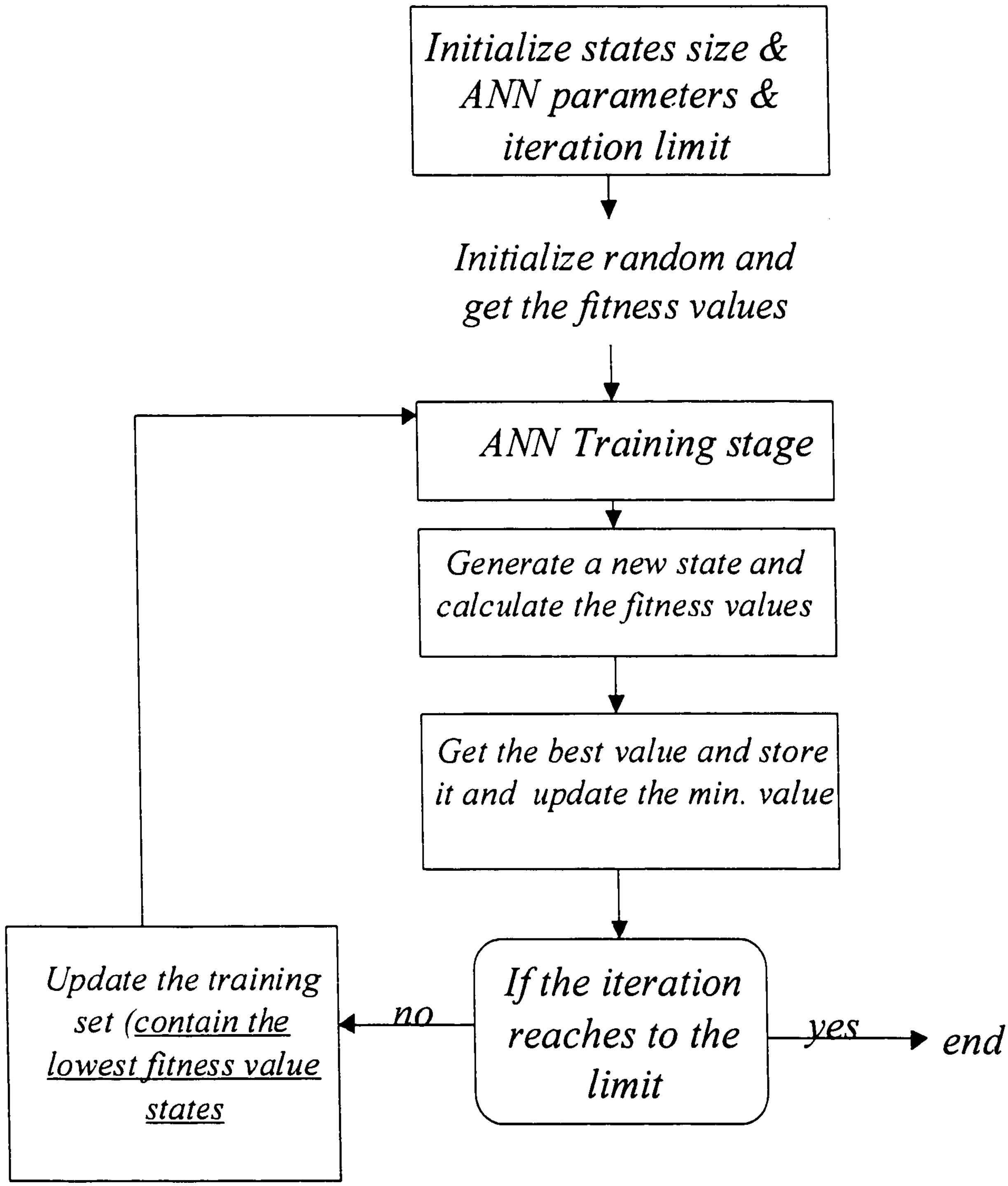


Figure 4.7 Flow chart of ANN solution procedures

4.5 Hybridization Methods with ANN

In this section, two hybridization methods are going to be discussed. The first one is the hybridization algorithm between the ANN and GA while the other one is the ANN hybridized with TS and GA.

4.5.1 ANN hybridized with GA

Since the ANN performs the local search, it will converge to the feasible solutions in the neighborhood of the initial state. If the initial state is set in the neighborhood of the optimal solution, the solution may be obtained. This is generally difficult as long as there is no information about the optimal solution. On the other hand, the GA forms a population based on a multi-point parallel search while escaping from local minimum points. This leads to improving the population by inheriting the good character of the parents to the offspring as useful information obtained in the past. Also, this leads to perform a global search and good solutions close to the optimal solution can be obtained. This means that if the initial population has information about the local minimum, solutions can be generated for a local search

From this discussion it can be seen that, if ANN and GA were hybridized, better results might be obtained. The ANN will be applied with the same searching strategy as in the last section but the best n-solution set will be stored. Then GA will be applied with the same strategy as in section 4.3 by considering the best n-solution set as an initial population. By implementing this method, it can be expected that the disadvantages of ANN and GA can be resolved as illustrated in Figure 4.8

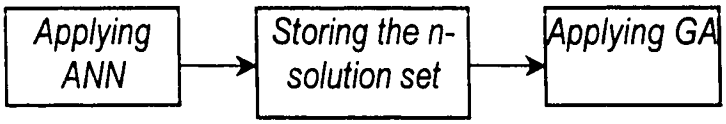


Figure 4.8 Hybridization algorithm between GA and ANN

4.5.2 Hybridizing Algorithm using TS, GA and ANN

Both TS and ANN perform a local search which leads them to state that the optimal solution is in the neighborhood of the starting solution set. This means that neither can guarantee obtaining the optimal solution. However, the GA has better performance since it can escape from the local to a global search. This section propose a hybridization algorithm using the good feature of ANN, GA and TS.

From section 4.4, the proposed algorithm was to apply GA operators and the Tabu list acceptance test through the random starting states. The currently proposed method does the same thing but the starting random states will be applied to ANN. The best n-solution obtained will be stored. Then the Tabu list acceptance test will be applied. The new state of solution set will be generated through the use of GA operators. The last two steps will be iterated until the optimal solution is obtained or the limit of the iteration is reached. Figure 4.9 shows the schematic diagram of the proposed method.

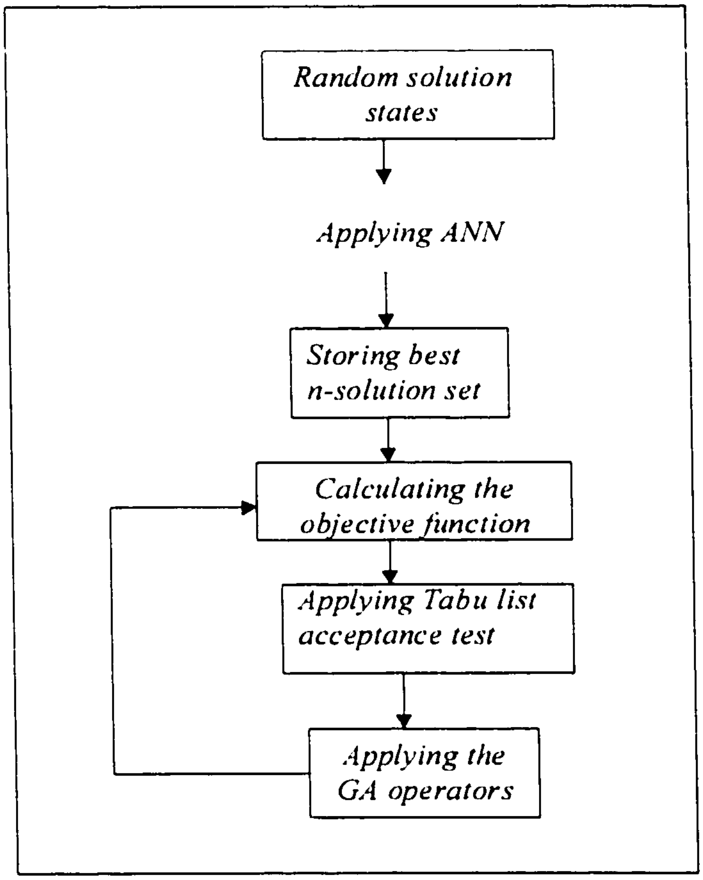


Figure 4.9 Hybridization method using ANN, TS and GA

CHAPTER – 5
SYSTEMS APPLICATIONS

The algorithms described in the last chapter will be used to solve the TEP of several power networks. The power networks include the Graver six-bus system, the IEEE-25 system and the SCECO-EAST future network.

5.1 *Simplified Network Expansion Model*

This section will demonstrate the solution obtained for a 6-bus system using Linear Programming and Quadratic Programming Models. It will also present the performance of TS, GA, ANN and their hybridization methods.

5.1.1 System Description

The Graver model (a six-bus system), as shown in Figure 5.1, is to be studied [9]. Buses 1 and 3 have both generation and load supplied: 2, 4, and are pure loads, and 6 is a new generation bus that needs to be connected to the network. The dotted lines represent possible line additions, and the solid lines are the existing lines. Table 5.1 shows the details of the six-bus system.

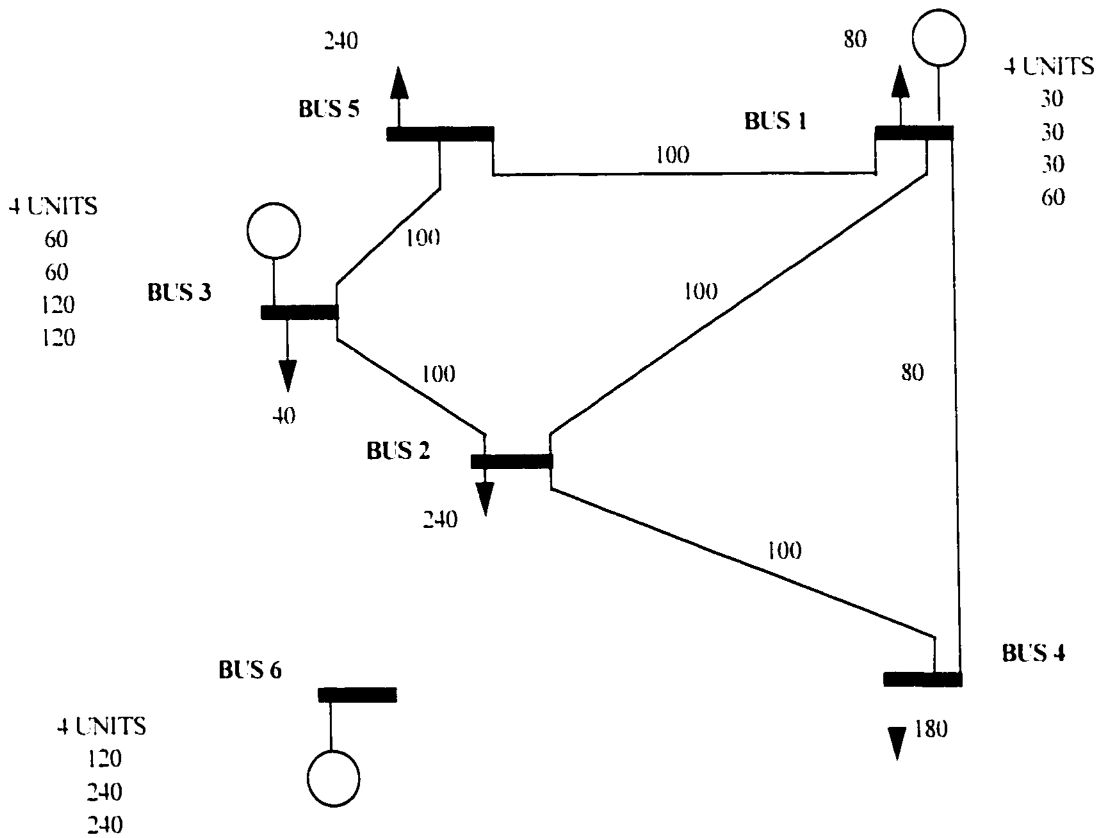


Figure 5.1 Initial six-bus system

When the system expands, the following is assumed:

1. *Four new possible Rights-of-Way: 2-6, 3-5, 4-6 and 5-6 are available*. This means that the total possible Rights-of-Way are nine as indicated in Table 5.1.

2. The connection between any two buses is allowed with a limit of *4 parallel paths in each right-of-way*. ($n_{ij}^{max} = 4$)

3. The maximum phase angle and the maximum difference of phase angle between the buses are assumed *as 20 degrees*. ($|\Delta\theta^{max}| = |\theta^{max}| = 20$ degrees)

4. The maximum power flow limit (P_{ij}^{max}) , susceptance (B_{ij}) with the cost (c_{ij}) of the new additional lines are listed in Table 5.1.

5. In the case of including the power losses to the objective function, the loss coefficient, K, is chosen to be 1000. Also, the P.U base in the DC-Load Flow analysis is 100MVA while the cost base is 10^5. The estimated life-time of the network lines is assumed to be 25 years and the cost of one kWh is assumed to be (0.005 monetary units).

Table 5.1 Six-Bus System Circuit Data

Bus: From - To	Right-of –way number	Cost (Units) (c_{ij})	Resistance (Ω) (R_{ij})	Suscept. ($1/\Omega$) (B_{ij})	Capacity (MW) (P^{max}_{ij})
1 - 2	1	40	0.100	2.50	100
1 - 4	2	60	0.150	1.67	80
1 - 5	3	20	0.050	5.00	100
2 - 3	4	20	0.050	5.00	100
2 - 4	5	40	0.100	2.50	100
2 - 6	6	30	0.075	3.33	100
3 - 5	7	20	0.050	5.00	100
4 - 6	8	30	0.075	3.33	100
5 - 6	9	61	0.120	1.64	78

Through the use of the proposed methods, the minimization algorithm is run recursively until there are no overloads in the system, i.e.,the problem is solved for testing the adequacy of a candidate solution; the adequacy is indicated by zero loss of load. Thus whenever a tentative solution set is inadequate, feasibility is achieved by the use of extra generators (loss of load). Since the objective function (equation (3.1)) has taken into account the effect of the power transmission cost; the line candidate with the largest power flow is the most effective in the expanded network.

5.1.2 Linear Programming Model Application:

The linear programming algorithm is introduced in this work for comparison purposes. This model is used because the non-linear part in the objective function (equation 3.1) is considered to be zero.

Before applying the model, the connection between the isolated bus (bus 6) and bus 2 is established. After running the DC load flow, it appears that there is a need for a 331 MW of circuit capacity between buses 6 and 4, 214 MW of circuit capacity between buses 6 and 2, and 61 MW of circuit capacity between buses 3 and 5 as shown in Figure 5.2.a. Figure 5.2.b shows the first addition, which is in the overload path between 6-4. The second addition following the largest overload path is the second line between the buses 6 and 4. The result is shown in Figure 5.2.c. Figure 5.2.d shows the new generator connection into the network by loop buses 2, 4, and 6. As a result of this construction, the 249 MW can be best delivered from bus 6 to bus 2. Figures 5.2.e and 5.2.f show the fourth and fifth additions in the same link. The 84 MW flow along link 3-5 constitutes the largest overload so an extra circuit is inserted as indicated in Figure 5.2.g. Finally, the last overload is along path 6-2 and a new circuit is planned there as shown in Figure 5.2.h. Table 5.2 summarizes the circuits added using a Linear Programming Model.

It appears that the optimal solution has a cost of 200 monetary units which is similar to the one provided by Graver with full generation scheduling (generator outputs 1, 3 & 6 are 50, 165 & 545 MW respectively). The additional circuits are as follows:

1. Four circuits between bus 2 and 6 (n_{26} = 4 circuits)
2. One circuit between bus 3 and 5 (n_{35} = 1 circuit)
3. Two circuits between bus 4 and 6 (n_{46}=2 circuit)

Figure 5.2.h shows the system after the expansion that represents the optimal solution.

Table 5.2 Iteration Additions Summary Using Linear Programming Model

Iteration #	***Location Of the addition***	***Corresponding Right-of-way***	***Reasons of the Addition***
1	*6 – 4*	*8*	*Reducing the largest overload path between 6 – 4 (331 MW)*
2	*6 – 4*	*8*	*Reducing the same overload path between 6 – 4 (231 MW)*
3	*6 – 2*	*6*	*New construction to connect the new generator to the network through the loop 2, 4, 6.*
4	*6 – 2*	*6*	*Reducing the largest overload path between 6 – 2 (249 MW)*
5	*6 – 2*	*6*	*Reducing the same overload path between 6 – 2 (149 MW)*
6	*3 – 5*	*–*	*Reducing the largest overload path between 3 – 5 (84 MW)*
–	*6 – 2*	*6*	*Reducing the last over load path between 6 – 2 (49 MW)*

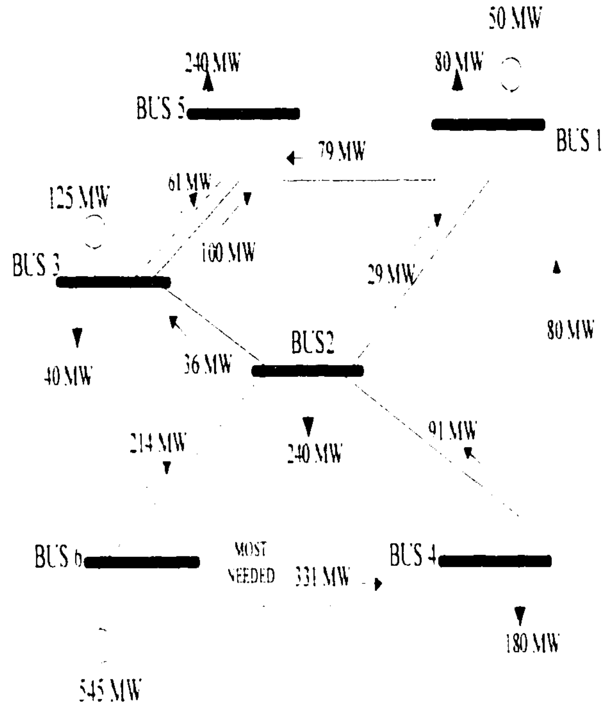


Fig.5.2.a Linear Programming Solution to The Six Bus Problem

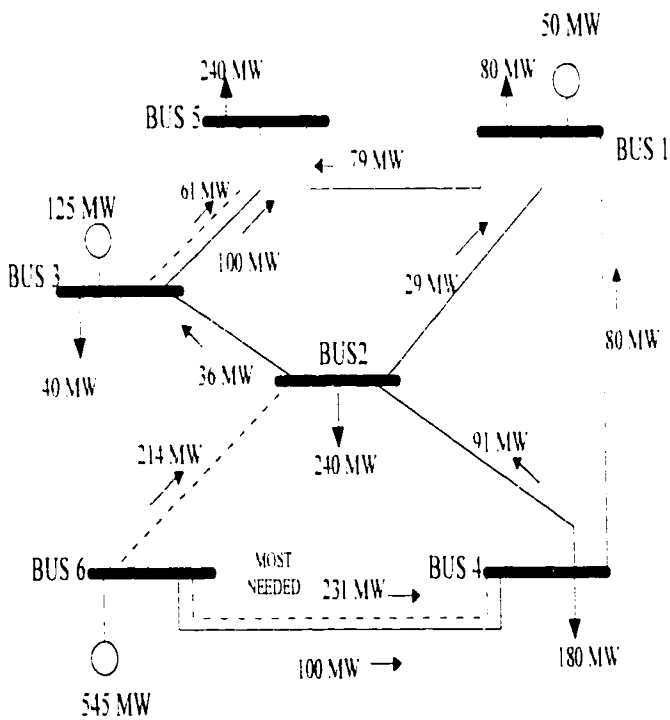


Fig. 5.2.b Solution after a First Circuit Addition

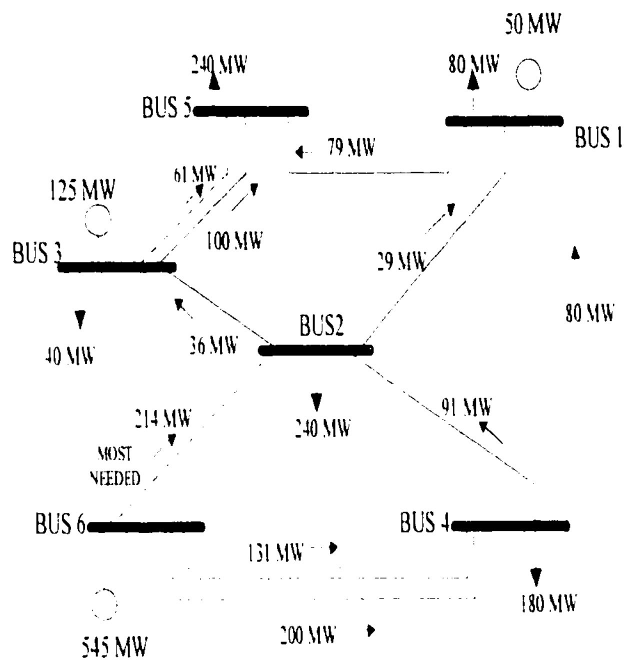


Fig.5.2.c Solution after a Second Circuit Addition

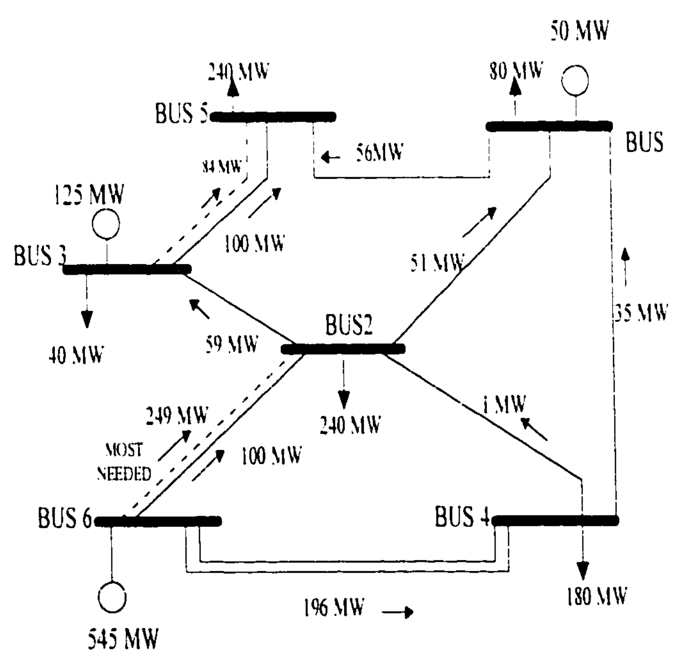


Fig. 5.2.d Solution after a Third Circuit Addition

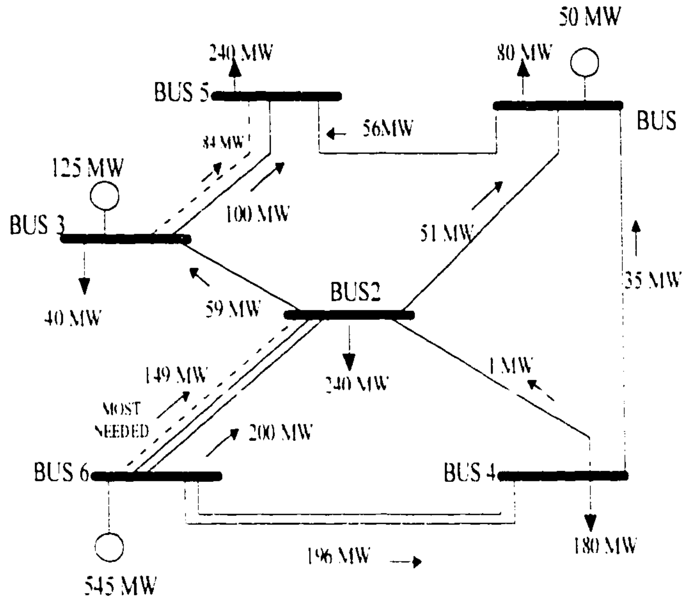


Fig 5.2.e Solution after a Fourth Circuit Addition

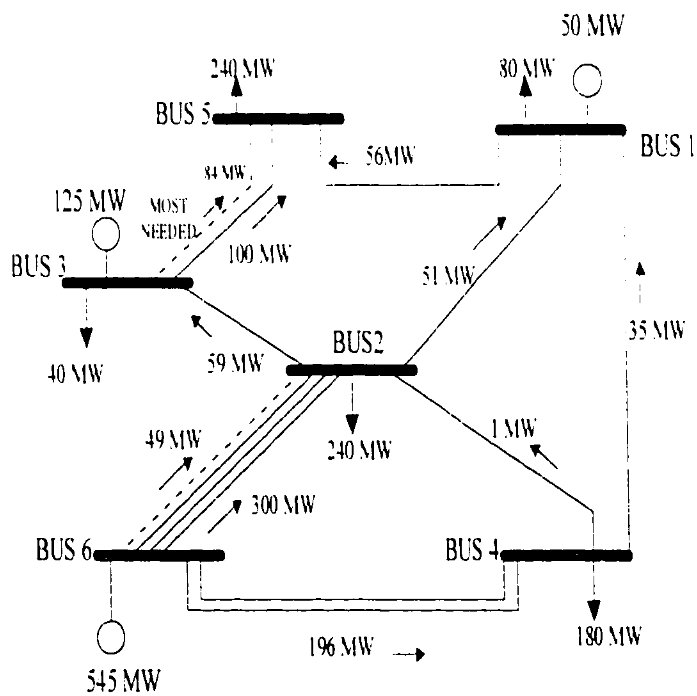


Fig. 5.2.f. Solution after a Fifth Circuit between Buses 6-2

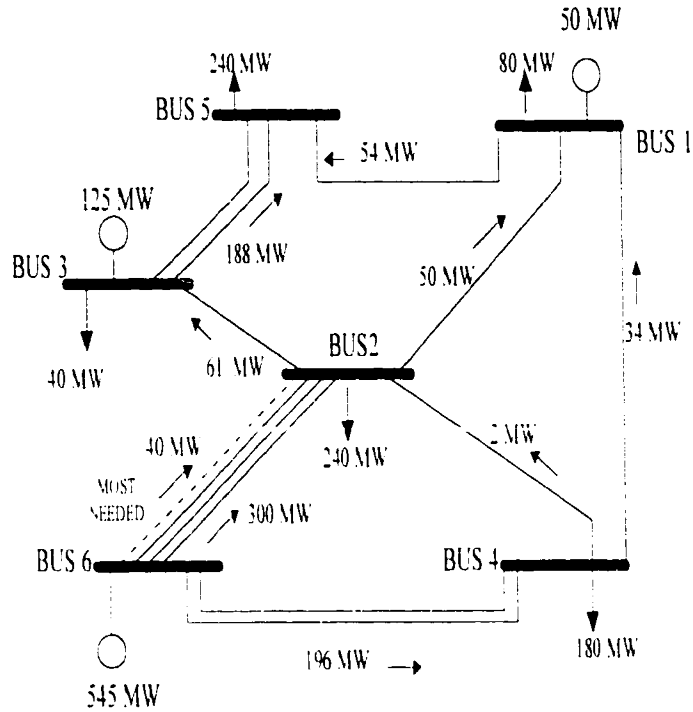


Fig. 5.2.g Solution after a Sixth Circuit Addition

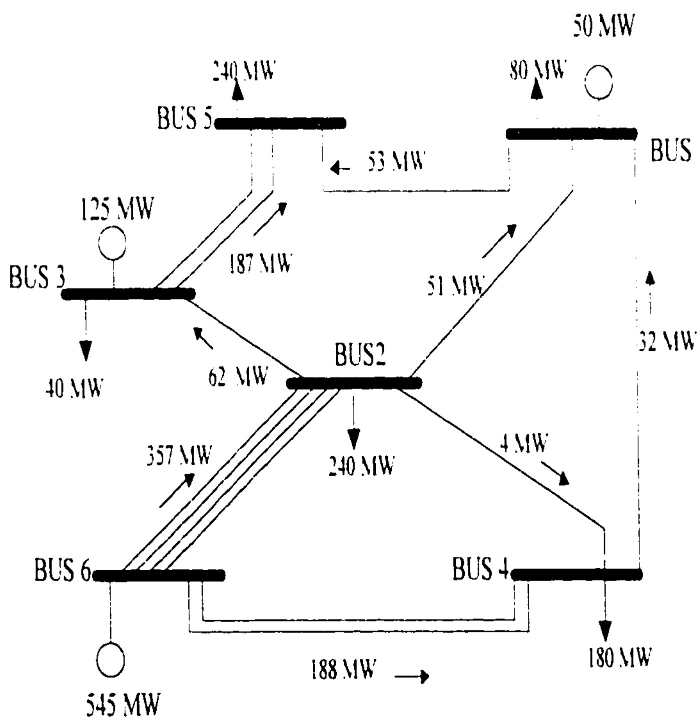


Fig. 5.2.h Solution after a Seventh Circuit Addition
Optimal Solution at K=0

5.1.3 Quadratic Programming Model

The quadratic programming method is also applied here for comparison purposes when the objective function includes the non-linear term element. The value of the coefficient of the power loss (K) is 1000.

The procedure of line addition that was used here is the same as in the Linear Programming Model. Through the test results of the model for the 6-bus, QP reaches to the optimal solution with total investment cost of 291 monetary units. Knowing that the ohmic power loss cost before the expansion of the transmission system is 1353.6 monetary units, the new additions lead to minimizing this cost to 382.54 monetary units. The new added lines are listed as the following:

1. Three circuits between bus 4 and bus 6
2. Four circuits between bus 2 and bus 6
3. One circuit between bus 5 and bus 6
4. One circuit between bus 3 and bus 5

Table 5.3 summarizes the line addition stages while Figure 5.3 shows the final network configuration.

Table 5.3 Additions Summary Using Quadratic Programming Model

Stage #	***Location Of the addition***	***Corresponding Right-of-way***	***Reasons of the Addition***
1	*6 – 4*	*8*	*Reducing the largest overload path between 6 – 4 (296 MW)*
	6 – 4	*8*	*Reducing the same overload path between 6 - 4 (196 MW)*
	6 – 2	*6*	*New construction to connect the new generator to the network through the loop 2, 4, 6.*
2	*6 – 4*	*8*	*Reducing the overload path between 6 – 4 (115 MW)*
	3 – 5	*7*	*Reducing the second overload path between 3 – 5 (73 MW)*
	6 – 2	*6*	*Reducing the third overload path between 6 – 2 (47 MW)*
	6 – 5	*9*	*Reducing the last over load path between 6 – 5 (21MW)*
3	*6 – 2*	*6*	*Reducing the largest overload path between 6 – 2 (19 MW)*
	6 – 4	*8*	*Reducing the last over load path between 6 – 4 (8MW)*

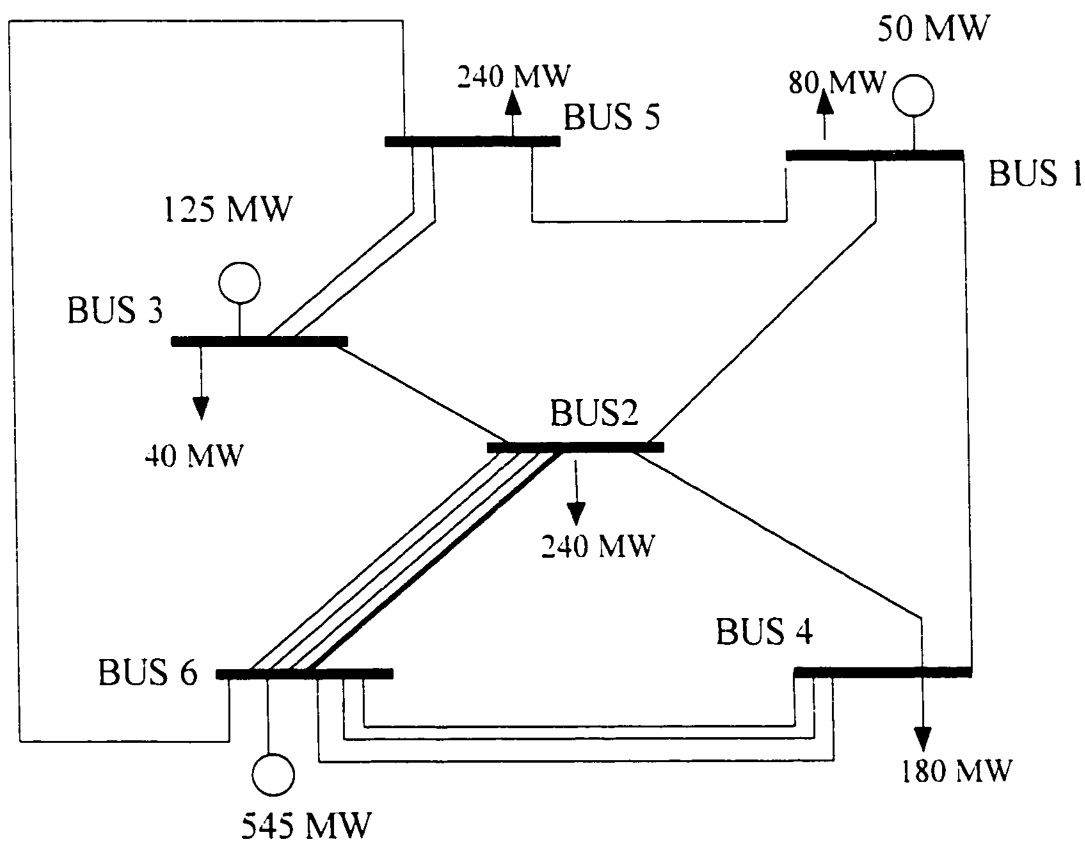


Figure 5.3 Optimal Solution With Adding the Non-linear Term to the Objective Function

5.1.4 Tabu Search Application

Through the application of the six-bus system, the TS operator's value settings are listed in Table 5.4.a. In addition, the representation code example of the TS for the six-bus system is shown in Table 5.4.b. The neighbor principle was applied by generating a new random movement such that a new state is a neighbor to an old state. A random number was generated and inserted in a random right-of-way. Also, a random right-of-way was selected to be detected by one for any random selected solution (new state).

TS converges to the optimal solution for both cases (at K=0 and 1000). It provides 200 monetary units when the cost of power losses was not considered and 291 when considering the power losses cost in the objective. Table 5.4.c shows the results of TS for the six-bus system. Figures 5.2.h and 5.3 shows the network scenarios after expansion for the two cases.

Table 5.4.a TS Setting Values

TS Operator	Setting Value
TS List Size	7
Movement Set	4
Total Iteration	1000
Vector Size	9

Table 5.4.b Representation Example of the Solution X Using TS.

Right-of-way number	1	2	3	4	5	6	7	8	9
Number of transmission lines	2	0	1	2	2	0	2	0	1

Table 5.4.c Results of TS for 6 – Bus System.

K=0
Investment Cost
200

K=1000			
Investment Cost	Power Losses Cost Before the New Lines Additions calculated for 25 years	Power Losses Cost After Expansion calculated for 25 years	Cost Saved By Minimizing the Power Losses calculated for 25 years
291	1353.6	382.45	971.15

5.1.5 Genetic Algorithm Application

During the use of the GA, the best values of its parameters were founded to be as follows: population size is assumed to be 15 and the chromosome length is 9. The crossover and mutation probabilities were set to be 0.82 and 0.07 respectively. GA was iterated until the maximum generation value of 1200 was reached. Table 5.5.a shows the GA operators setting values while Table 5.5.b shows the chromosome code formulation example for the GA.

The performance of GA and TS is the same because both converge to the optimal solution for both cases (at K= 0 and 1000). They provide 200 monetary units when the cost of power losses was excluded and 291 under the consideration of power loss cost. Moreover, Table 5.5.c shows the detailed results of GA for the six-bus system.

Table 5.5.a GA Setting Values

GA Operator	Setting Value
Population Size	15
Crossover Probability	0.82
Mutation Probability	0.07
Chromosome Size	9
Total Generation	1200

Table 5.5.b Chromosomes Coding Example for GA of the Six-Bus System.

Locus = right-of-way number	1	2	3	4	5	6	7	8	9
Gene-number of transmission lines	2	1	3	0	0	2	0	1	2

Chromosome: connection state of transmission lines in the power system.

Table 5.5.c Results of GA for 6 – Bus System.

K=0
Investment Cost
200

K=1000			
Investment Cost	Power Losses Cost Before the New Lines Additions calculated for 25 years	Power Losses Cost After Expansion calculated for 25 years	Cost Saved By Minimizing the Power Losses calculated for 25 years
291	1353.6	382.45	971.15

5.1.6 Hybridization Algorithm between TS and GA Application:

The hybridization of GA & TS will be used to solve the TEP of the 6-bus system. It is expected that this will increase the ability of obtaining the optimal solution. This is because the tracking algorithm of the two methods is a combination of GA and TS as explained in section 4.3. Table 5.6.a shows the setting values of the parameters for both proposed algorithms. By running both for the same system, they obtained the same optimal solution for both cases. This means they reach to 200 monetary units when K=0 and to 291 monetary units when K=1000. Table 5.6.b shows the results obtained by both methods for the same system.

Table 5.6.a Proposed Method Parameters

Model	*The Parameter*	*Setting Value of model one*	*Setting Value of model two*
TS	TS List Size	7	7
	Movement Set	4	N/A
	Total Iteration	1000	N/A
	Vector Size	9	9
GA	Population Size	15	15
	Crossover Probability	0.82	0.82
	Mutation Probability	0.07	0.07
	Chromosome Length	9	9
	Total Generation	1200	1200

Table 5.6.b Results Obtained for 6 – Bus System.

K=0
Investment Cost
200

K=1000			
Investment Cost	Power Losses Cost Before the New Lines Additions *calculated for 25 years*	Power Losses Cost After Expansion *calculated for 25 years*	Cost Saved By Minimizing the Power Losses *calculated for 25 years*
291	1353.6	382.45	971.15

5.1.7 Artificial Neural Network Application

The parameter of the ANN that was set to solve the six bus system are listed in Table 5.7.a while Table 5.7.b shows the code representation example of the solution using ANN.

The ANN algorithm was not able to reach the optimum solution as shown in Table 5.7.c. It converged to a reasonable solution with a cost of 231 monetary units at K=0 with the following circuit additions as shown in Figure 5.4.a:

1. Three circuits between bus 2 and bus 6
2. Two circuits between bus 4 and bus 6
3. One circuit between bus 5 and bus 6
4. One circuit between bus 3 and bus 5

Also, at K= 1000, it reaches to 261 monetary units with the following additions:

1. Three circuits between bus 2 and bus 6
2. Three circuits between bus 4 and bus 6
3. One circuit between bus 5 and bus 6
4. One circuit between bus 3 and bus 5

Figure 5.4.b shows the resultant network.

Table 5.7.a ANN Setting Values

ANN Parameter	Setting Value
Neurons of the Input Layer	1
Neurons of the Hidden Layer	3
Neurons of the Output Layer	9
The Training Set at Each Iteration	50
Training Error	0.08
Total Number of Iterations	2000

Table 5.7.b Representation Example of the Solution States Using ANN

Output neuron number	1	2	3	4	5	6	7	8	9
Right-of-way number	1	2	3	4	5	6	7	8	9
Number of transmission lines	2	0	1	2	2	0	2	0	1

Table 5.7.c Summary Performance of ANN

K=0
Investment Cost
231

K=1000			
Investment Cost	Power Losses Cost Before the New Line Additions *calculated for 25 years*	Power Loss Cost After Expansion *calculated for 25 years*	Cost Saved By Minimizing the Power Losses *calculated for 25 years*
261	1353.6	448.83	904.77

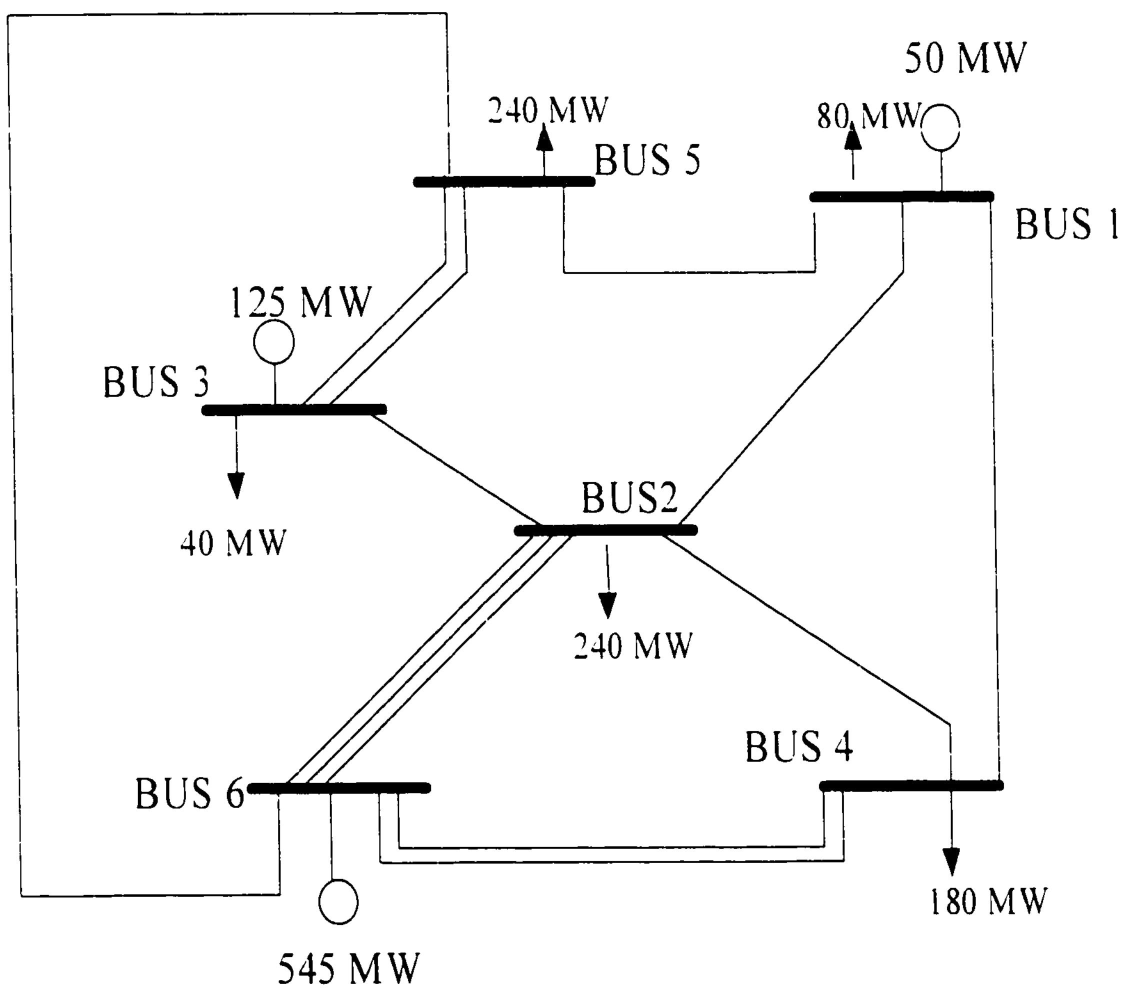


Figure 5.4.a Optimal solution using ANN without adding the non-linear term to the objective function

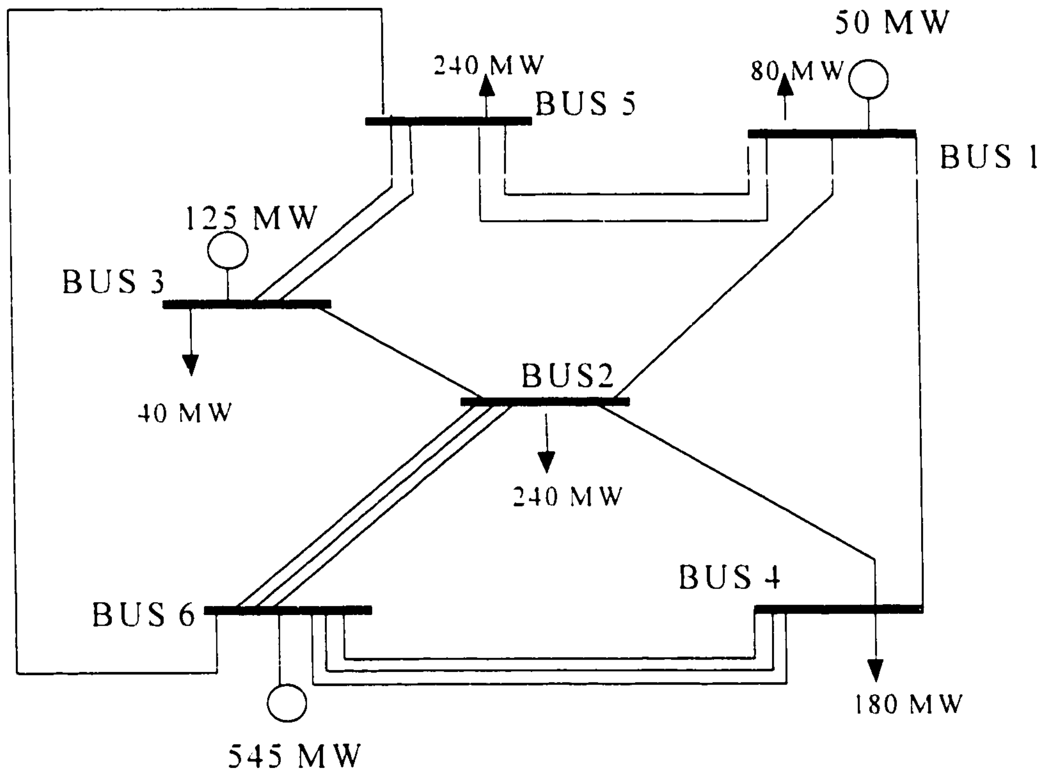


Figure 5.4.b Optimal solution using ANN with adding the non-linear term to the objective function

5.1.8 Application of the hybridization algorithms with ANN

The TEP of the six-bus system is solved by using two hybrid algorithms with ANN. The first one is ANN with the GA method as described in section 4.5 while the second one is the hybridization algorithm between TS, ANN and GA. Tables 5.8 and 5.9 show the method setting values. Through the test results, it shows the same result that was obtained using TS or GA when they were applied individually (200 monetary units at K=0 and 291 monetary units at K=1000). The main feature that can be obtained from these algorithms is that it can increase the probability of reaching the optimal solution. This is because both TS and ANN perform good local search while GA has the ability to escape from the local search to a global optimal search. Table 5.10 shows the additional line costs for both cases.

Table 5.8 Proposed Method Parameters

Model	The Parameter	Setting Value
ANN	Neurons of the Input Layer	1
	Neurons of the Hidden Layer	3
	Neurons of the Output Layer	9
	The Training Set at Each Iteration	50
	Training Error	0.08
	Total Number of Iteration	2000
GA	Population Size	15
	Crossover Probability	0.82
	Mutation Probability	0.070
	Chromosome Length	9
	Total Generation	1200

Table 5.9 Proposed Method Parameters

Model	The Parameter	Setting Value
ANN	Neurons of the Input Layer	1
	Neurons of the Hidden Layer	3
	Neurons of the output Layer	9
	The Training Set at Each Iteration	50
	Training Error	0.08
	Total Number of Iteration	2000
TS	TS List Size	7
	Vector Size	9
GA	Population Size	15
	Crossover Probability	0.82
	Mutation Probability	0.07
	Chromosome Length	9
	Total Generation	1200

Table 5.10 Results Obtained for 6 – Bus System.

K=0
Investment Cost
200

K=1000			
Investment Cost	Power Losses Cost Before the New Lines Additions calculated for 25 years	Power Losses Cost After Expansion calculated for 25 years	Cost Saved By Minimizing the Power Losses calculated for 25 years
291	1353.6	382.45	971.15

5.1.9 Summary of the Artificial Intelligent (AI) Algorithms

In summary, through the application of the 6-bus system, all AI and classical models (Linear and Quadratic Programming methods) can search and obtain the optimal solution except the ANN. The ANN recorded a higher cost in the case of excluding the cost of power losses (K=0) and the lowest saved cost of the ohmic power losses when K=1000. This is because the initial states of ANN are not set in the neighborhood of the optimal solution. This means that the convergence is very difficult as long as there is no information about the optimal solution. Although TS and GA cannot guarantee the optimal solution, they converged to the optimal one. This is because they do not stick with the local minimum during the search. Table 5.11 summarizes the performance of all methods that have been applied to the six- bus system.

Table 5.11 Summary Results of Six – Bus System by Applying All Proposed Methods

K=0	
Model	*Investment Cost*
TS	200
GA	200
ANN	231
GA – TS (1)	200
GA – GA (2)	200
ANN – GA	200
ANN- TS- GA	200
Linear Model	200

K=1000			
Model	*Investment Cost*	*Losses Cost after the new line additions(*)* (calculated for 25 years)	*Cost saved by minimizing the power losses (**)* (calculated for 25 years)
TS	291	382.54	971.06
GA	291	382.54	971.06
ANN	261	448.83	904.77
GA-TS (1)	291	382.54	971.06
GA-TS (2)	291	382.54	971.06
ANN- GA	291	382.54	971.06
ANN- TS- GA	291	382.54	971.06
Quadratic Model	291	382.54	971.06

(**) This cost is calculated as difference between the cost of ohmic power losses before the new lines additions (1,353.6 monetary units) based on 25 years line's life-time and the cost calculated after the expansion as in (*) for the same period.

5.1.10 Right-of-way Constraints Study

To complete the study on the 6-bus system, the validity of right-of-way is applied. The reason of this study is to test the ability of any of AI to obtain the optimal solution if the right-of-ways are limited. Using the method that hybridizes the TS, GA and ANN, the results show that when the right-of-way between bus 3 and 5 is not allowed, the optimal cost obtained when K=0 is 251 monetary units, as shown in Figure 5.5. While, at K=1000, the total investment costs of the new additions are 332 monetary units. In this case, the cost that could be saved though this addition for 25 years is 968.93 monetary units. Figure 5.6 shows the locations and the quantity of the new additions. In the second case when the right-of-way between bus 2 and 6 is not valid the cost reaches to 334 monetary units as shown in Figure 5.7 but when the power loss cost was considered in the calculation, the total investment cost will be 384 monetary units. Figure 5.8 shows the new network that can save 811.63 monetary units over 25 years. The final case is that the right-of-way between bus 6 and 4 is not allowed. The method reaches to 282 monetary units as indicated in Figure 5.9 without including the power loss cost consideration. Also, Figure 5.10 shows the scenario of the new additions that costs 343 monetary units when K equals 1000. The cost that can be saved for 25 years though this addition is 816.27 monetary units. Table 5.12 shows the results of the 6- bus system when right-of-way validity is applied.

Table 5.12 Rights-of-Way Study Results Using the Hybridizing Algorithm between GA, TS and ANN

K=0		
Case #	*Description*	*Investment cost*
1	*Right-of-way 3 – 5 is not allowed*	251
2	*Right-of-way 2 – 6 is not allowed*	334
3	*Right-of-way 4 – 6 is not allowed*	282

K=1000				
Case #	*Description*	*Investment Cost*	*Losses Cost after the new line additions(*)* *(calculated for 25 years)*	*Cost saved by minimizing the power losses(**)* *(calculated for 25 years)*
1	*Right-of-way 3 – 5 is not allowed*	332	348.67	968.93
2	*Right-of-way 2 – 6 is not allowed*	384	541.97	811.63
3	*Right-of-way 4 – 6 is not allowed*	343	537.33	816.27

(**) This cost is calculated as difference between the cost of ohmic power losses before the new lines additions (1,353.6 monetary units) based on 25 years line's life-time and the cost calculated after the expansion as in (*) for the same period.

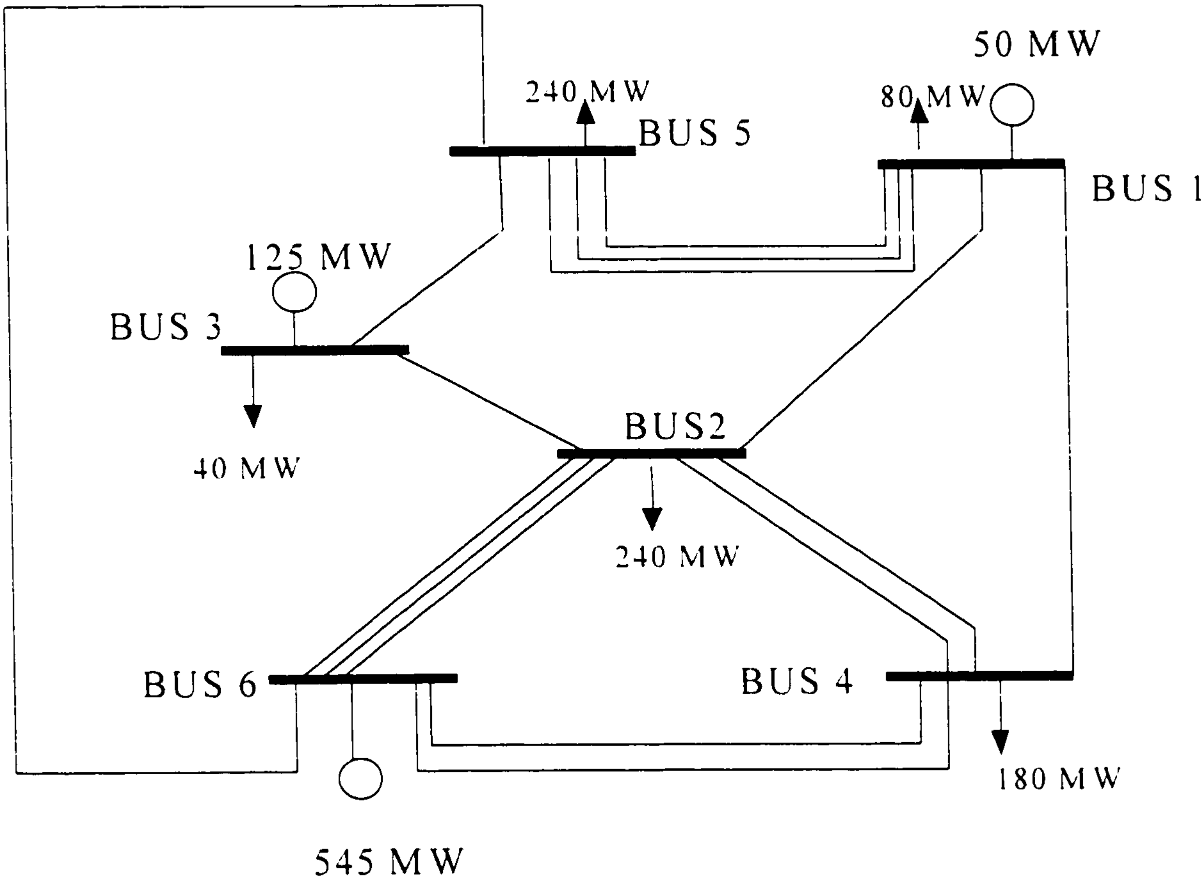


Figure 5.5 Optimal solution without adding the non-linear term to the objective function when the right-of-way 3 –5 is not allowed

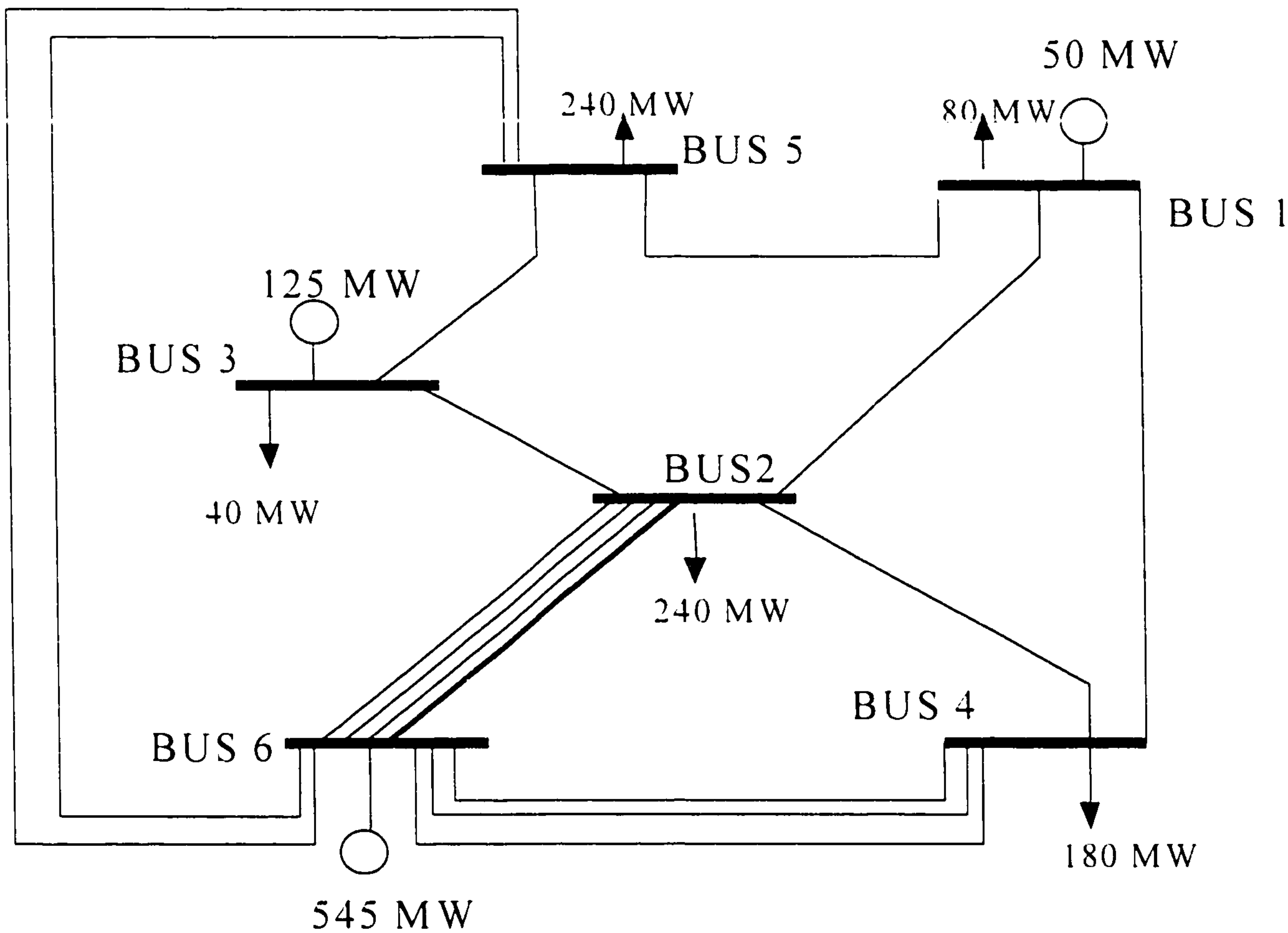


Figure 5.6 Optimal solution with adding the non-linear term to the objective function when the right-of-way 3 – 5 is not allowed

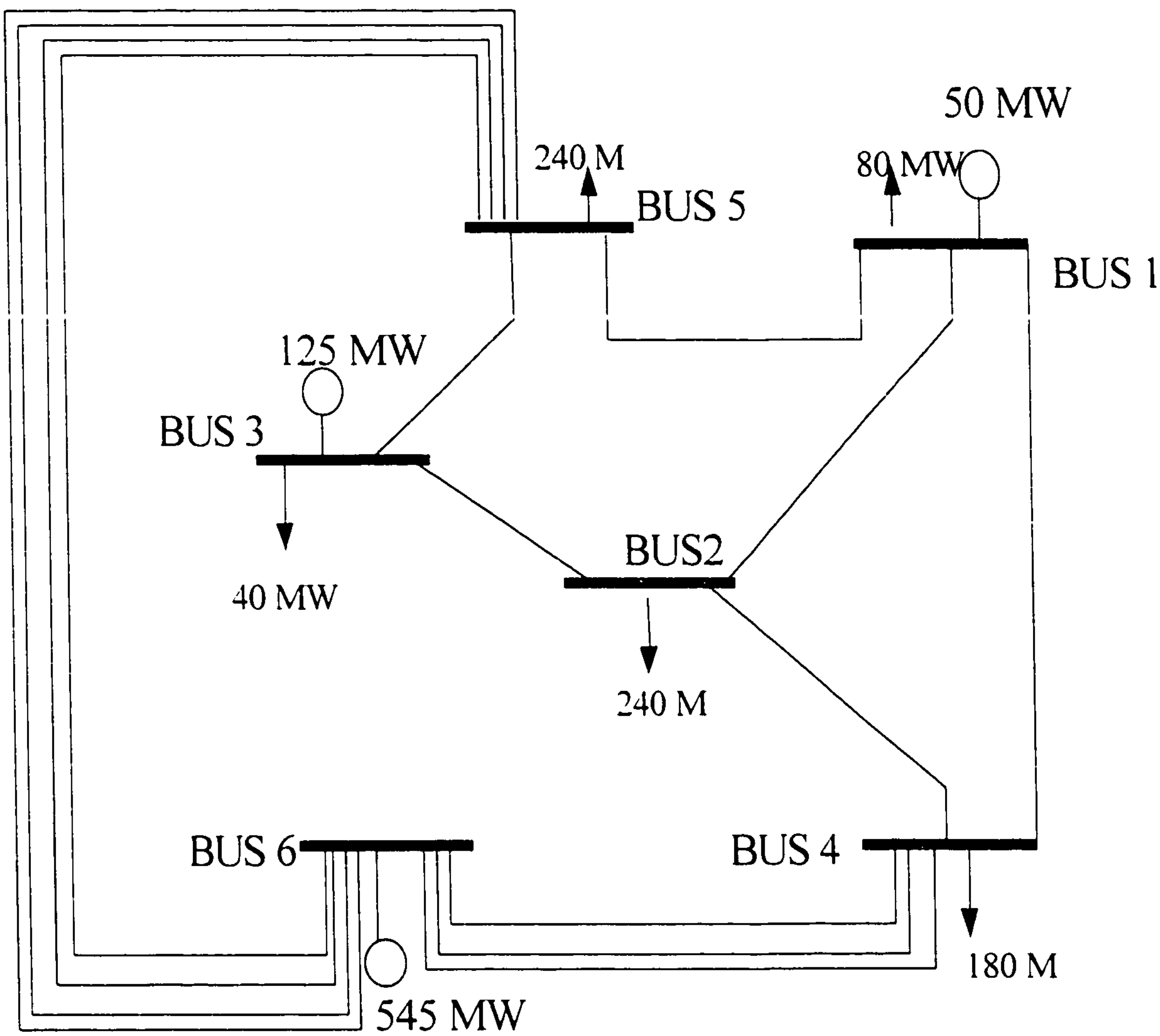


Figure 5.7 Optimal solution without adding the non-linear term to the objective function when the right-of-way 2 – 6 is not allowed

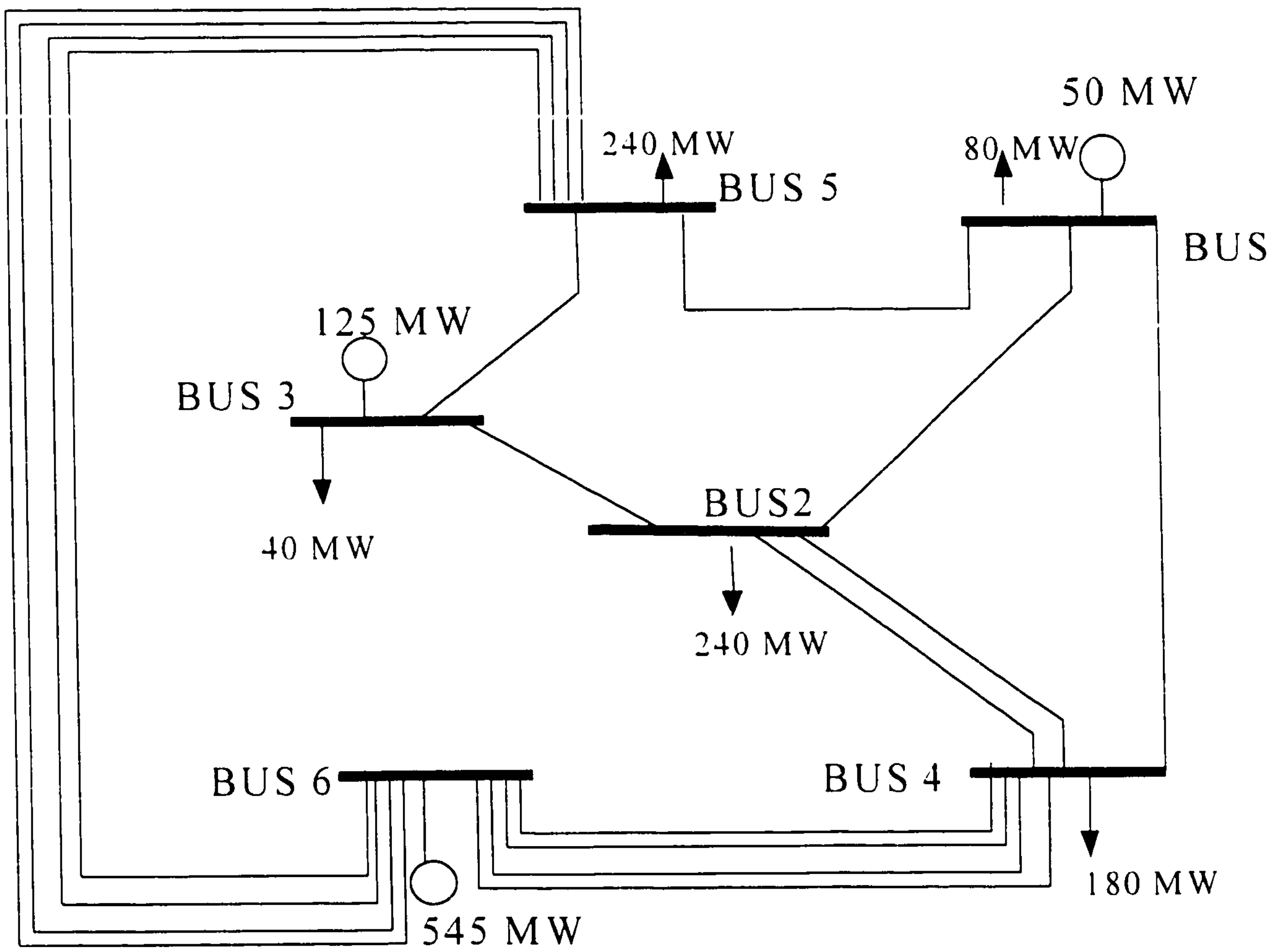


Figure 5.8 Optimal solution with adding the non-linear term to the objective function when the right-of-way 2 – 6 is not allowed

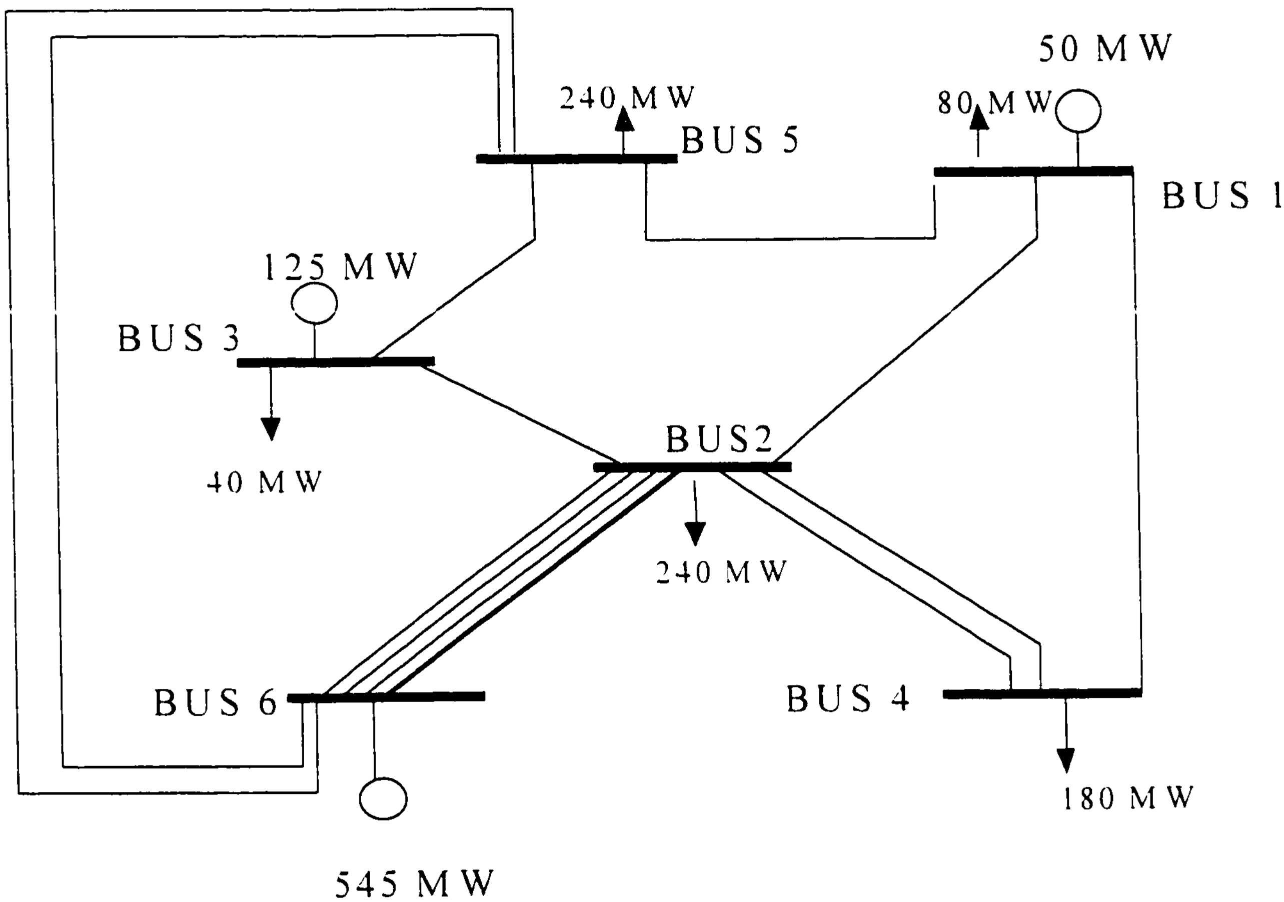


Figure 5.9 optimal solution without adding the non-linear term to the objective function when the right-of-way 4 – 6 is not allowed

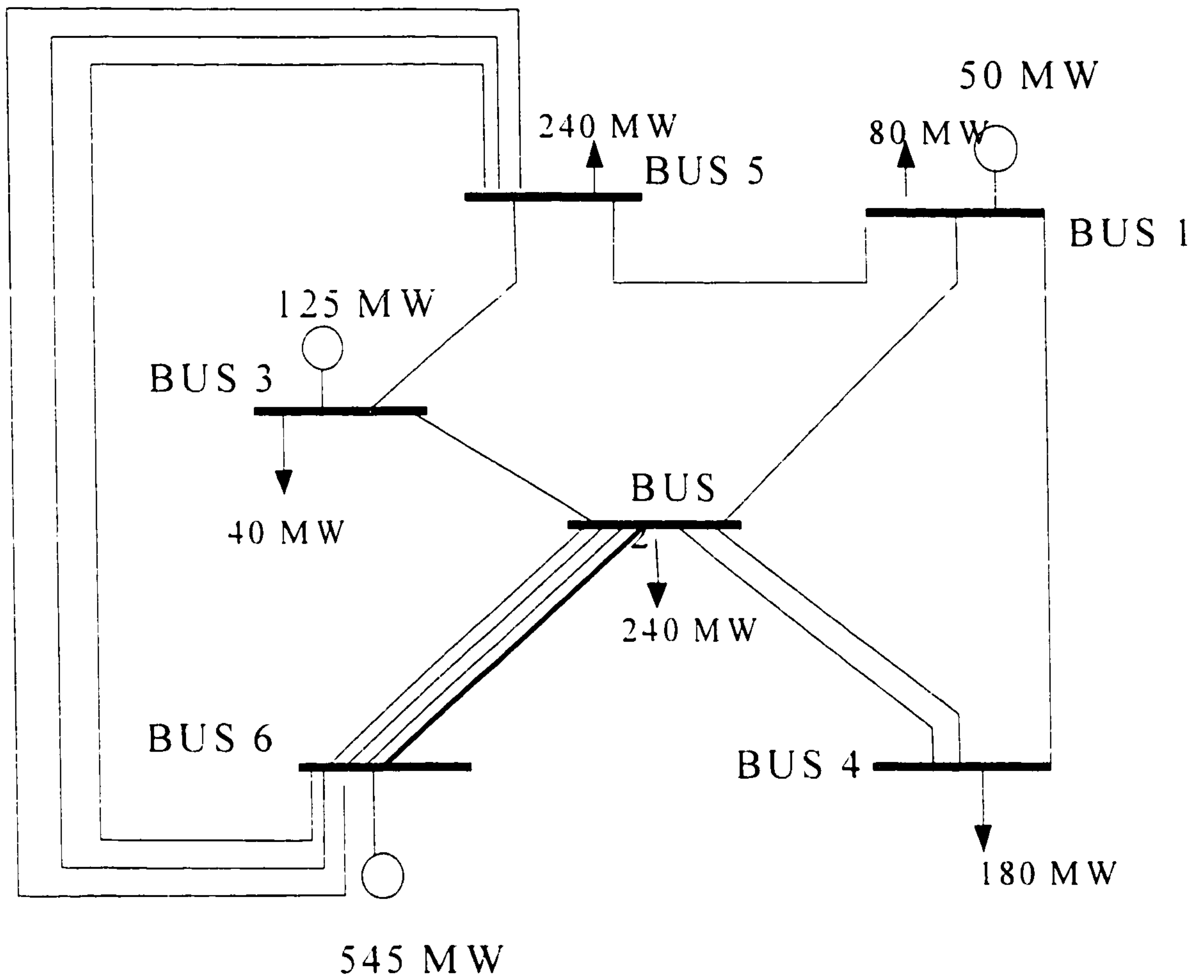


Figure 5.10 Optimal solution with adding the non-linear term to the objective function when the right-of-way 4 – 6 is not allowed

5.2 IEEE - 25 bus system

This section will present a complete study of the performance of TS, GA, ANN and their hybridization methods via the classical methods (QP, LP) to solve the static TEP through the application of the IEEE – 25 bus system.

5.2.1 System Description

The IEEE-25 bus, as shown in Figure 5.11, is to be used [42]. This system will be studied under the following assumptions:

i) The loads at all nodes are increased by 10% per annum and at a future period a new generation plant is to be built to satisfy the increased load.

ii) Reactance of the new circuits is going to be chosen from Table 5.13 and a *maximum of 4 extra lines are* allowed in each of *the 36 possible rights-of-way with equal importance of all transmission lines* ($n_{ij}^{max} = 4$).

iii) The maximum phase angle and the maximum difference of phase angle between the buses are assumed *as 20 degrees*. ($|\Delta\theta^{max}| = |\theta^{max}| = 20$ degrees)

iv) The maximum power flow limit (P_{ij}^{max}) and the reactance (X_{ij}) with the cost (c_{ij}) of the new additional lines are listed in Table 5.13.

v) In the case of including the power losses in the objective function, the loss coefficient, K, is chosen to be 10000. This means that the estimated life time of the network lines is assumed to be 25 years while the cost of one kWh is assumed to be (0.042 SR/kWh). Also, the P.U base in the DC- Load Flow analysis is 100MVA and the base of the lines to be added is 10^4.

The dotted lines represent possible line additions while the solid lines are the existing lines. The buses are numbered from 1 to 25. The new bus is 25, connected to buses 5 and 24. The load and generation data can be found in Tables 5.13 and 5.14.

During the analysis of this system, the factors that affect expansion were considered. The economic goal that aims to avoid the construction of a 230/138KV transformer station and, instead, adds the lines to the existing 230KV or 138KV substations is the first factor. The second factor is the technical aspects. This factor deals with the thermal rating of the added lines and the phase difference between the buses.

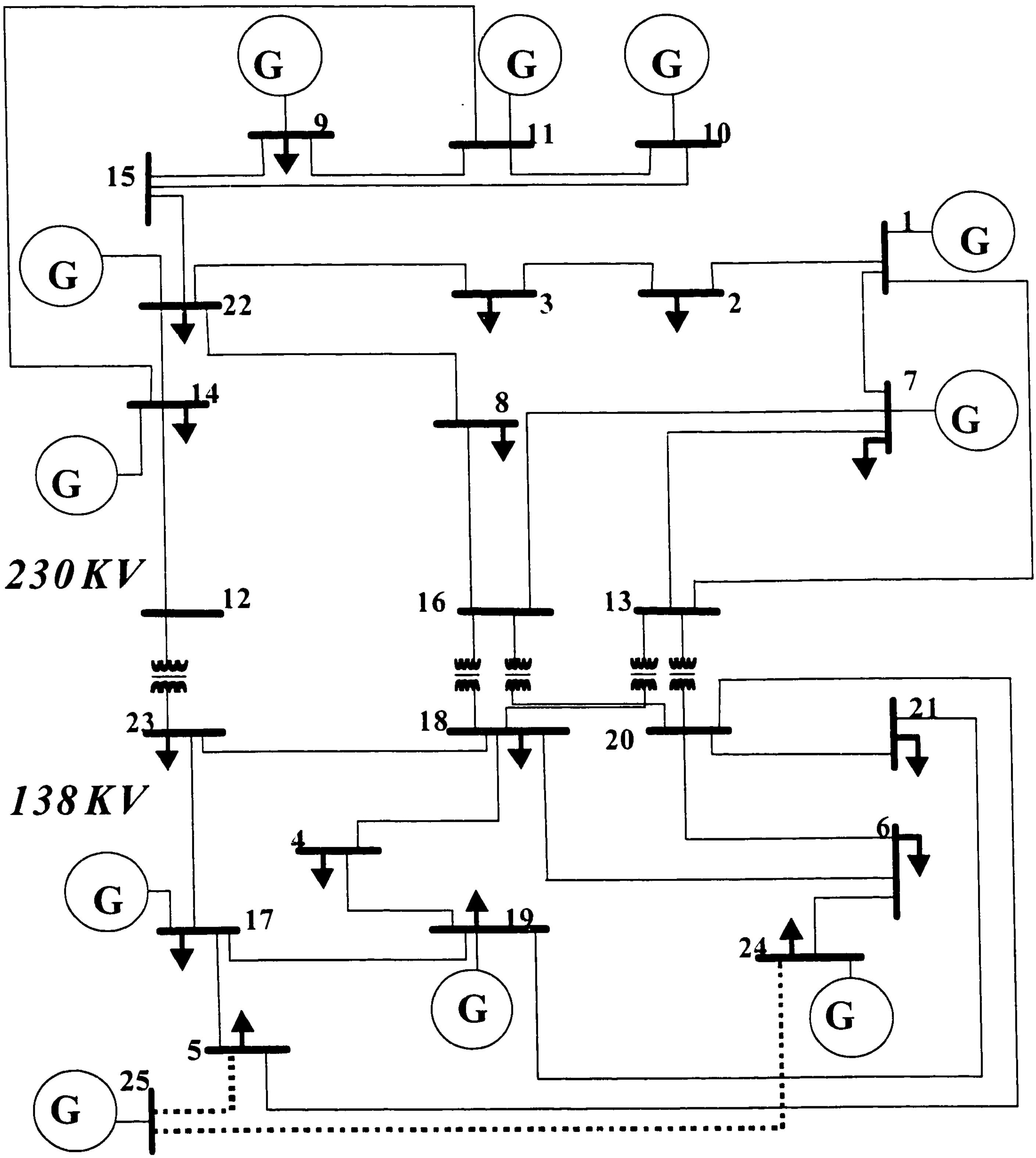


Figure 5.11 IEEE – 25 Bus Network

Table 5.13 Line Bus Data

Rights-of-Way No.	*Line Link Bus From – To*	*Resistance (Ω)*	*Reactance (Ω)*	*Capacity (P.U.)*	*Distance (KM)*	*Voltage level (KV)*	*Cost X10000 SR*
1	1 – 2	0.0027	0.0108	8.0	16.40	230	1410
2	1 – 7	0.0217	0.0865	0.65	132.4	230	10428
3	1 – 13	0.0236	0.0966	1.00	147.1	230	11613
4	2 – 3	0.0050	0.0198	5.00	30.70	230	2666
5	3 – 22	0.0058	0.0231	2.00	35.70	230	3070
6	4 – 18	0.0260	0.1037	10.00	39.83	138	1840
7	4 – 19	0.0316	0.1267	2.50	48.73	138	2240
8	5 – 17	0.0212	0.0854	8.00	32.50	138	1495
9	5 – 20	0.0221	0.0883	9.40	33.87	138	1564
10	6 – 18	0.0417	0.1651	4.40	63.44	138	2916
11	6 – 20	0.0417	0.1651	2.80	63.44	138	2916
12	6 – 24	0.0153	0.0614	10.80	23.62	138	1104
13	7 – 13	0.0119	0.0476	2.50	72.60	230	6235
14	7 – 16	0.0119	0.0476	0.90	72.60	230	6235
15	8 – 16	0.0104	0.0418	4.90	64.50	230	5547
16	8 – 22	0.0097	0.0389	0.65	60.20	230	5160
17	9 – 11	0.0032	0.0129	2.60	20.30	230	1720
18	9 – 15	0.0036	0.0144	2.50	22.30	230	1917
19	10 – 11	0.0169	0.0678	8.00	103.60	230	8216
20	10 – 15	0.0259	0.1053	2.50	161.10	230	12720
21	11 – 14	0.0061	0.0245	7.00	37.10	230	3190
22	12 – 14	0.0130	0.0519	1.00	80.30	230	6343
23	12 – 23	0.0210	0.0839	0.70	5.30	138	253
24	13 – 18	0.0210	0.0839	1.00	5.30	138	253
25	13 – 20	0.0210	0.0839	2.50	5.30	138	253
26	14 – 22	0.0043	0.0173	2.00	26.80	230	2236
27	15 – 22	0.0065	0.0259	3.60	40.32	230	3466
28	16 – 18	0.0210	0.0839	2.50	5.30	138	253
29	16 – 20	0.0210	0.0839	5.64	5.30	138	253
30	17 – 19	0.0035	0.0139	4.00	4.00	138	185
31	17 – 23	0.0532	0.2112	3.50	81.20	138	3330
32	18 – 23	0.0298	0.1190	1.50	45.77	138	2102
33	19 – 21	0.0418	0.1920	1.10	73.84	138	3392
34	20 – 21	0.0152	0.0605	1.80	18.30	138	842
35	5 – 25	0.0451	0.0902	2.20	14.00	138	649
36	24 – 25	0.0226	0.1805	2.20	25.00	138	1150

Table 5.14 Generation and Load Data

Bus No	*Generation Power (p.u)*	*Minimum Capacity (p.u)*	*Maximum Capacity (p.u)*	*Real Power Demand (p.u)*
1	5.30	1.32	6.60	-
2	-	-	-	1.28
3	-	-	-	1.81
4	-	-	-	0.74
5	-	-	-	0.71
6	-	-	-	0.71
7	5.94	1.19	5.95	2.65
8	-	-	-	1.94
9	4.00	0.8	4.00	3.33
10	3.00	0.6	3.00	-
11	4.00	0.8	4.00	-
14	0.43	0.43	2.15	3.17
17	0.40	0.38	1.92	1.08
18	-	-	-	1.75
19	0.40	0.38	1.92	0.97
20	-	-	-	1.95
21	-	-	-	1.36
22	1.55	0.31	1.55	1.00
23	-	-	-	1.80
24	0.60	0.60	3.00	1.25
25	3.30	1.32	6.60	-

5.2.2 Application of Linear (LP) and Quadratic Programming (QP) models:

The classical optimization methods (LP and QP) were applied to the IEEE – 25 bus system. LP was used when the objective function was a linear while the QP was applied in case of having a non-linear term.

For K = 0, LP obtained a total investment cost SR million 638.25 but, at K = 10000, the QP reaches to SR million 680.47 as cost of the new additions. Knowing that the calculated cost of the ohmic power losses of the network before adding the new lines, for 25 years, is SR million 1,162.40, the results of QP minimize this cost to SR million 741.590. Table 5.15 shows the results this cost in details while tables 5.16.a and 5.16.b show the network configuration for the two cases.

Table 5.15 Summary Performance of LP and QP

K=0 (LP)
Total cost SR million
638.250

K=10000 (QP)			
Investment cost SR million	Ohmic Losses Cost Before the New Lines Additions SR million (Calculated for 25 years)	Ohmic Losses Cost After the Expansion SR million (Calculated for 25 years)	Saved Cost By Minimizing Ohmic Power Losses Through the Expansion (Calculated for 25 years)
680.470	1,162.40	741.590	420.80

Table 5.16.a Configuration Results Using LP
Cost is SR million 638.25

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	2	1 – 7	1
2	3	1 – 13	1
3	14	7 – 16	3
4	16	8 – 22	2
5	22	12 – 14	3
6	23	12 – 23	1
7	24	13 – 18	1
8	35	24 – 25	1
9	36	5 – 25	1

Table 5.16.b Configuration Results Using QP
Cost is SR million 680.470

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	2
2	5	3 – 22	1
3	14	7 – 16	2
4	16	8 – 22	3
5	22	12 – 14	2
6	23	12 – 23	3
7	24	13 – 18	2
8	34	20 – 21	1
9	35	24 – 25	2
10	36	5 – 25	2

5.2.3 Application of TS, GA and their hybridization methods

By applying TS, GA and their hybridization methods to the system, the parameters of each are defined in Tables 5.17 and 5.18, respectively, where as the performance results are summarized in Table 5.19.

At K = 0, the GA has better performance in terms of cost values when compared to the TS. It obtains (SR) million 554.990 while the TS converges to the network, which costs SR million 555.550. The hybridization methods improve the cost to (SR) million 548.070 for the first algorithm and (SR) million 535.820 for the second method. The best configurations obtained by each method are listed in Tables 5.20.a, 5.20.c, 5.20.e and 5.20.g.

On the other hand, when the cost of power losses is considered in the calculation in which K equals to 10000, the GA still has better performance compared to TS. It converges to SR million 609.490 as an investment cost and it minimizes the cost of the ohmic power losses to SR million 644.80 which means it can save SR million 517.56. Although TS minimizes the cost of the ohmic power losses to better values than the GA (SR million 582.240), its network configuration costs SR million 677.490 which is more than the cost of GA. Also, The hybridization methods minimize the ohmic power losses to SR million 582.72 for model–1 and to SR million 591.920 for model–2, though their investment cost is greater than GA, SR million 655.130 and SR million 643.630 for the two models respectively. This means model – 1 saved SR million 579.68 and model – 2 saved SR million 570.48 compared to the cost of the ohmic power losses before the expansion. The best configurations obtained by each method are listed in Tables 5.20.b, 5.20.d, 5.20.f, and 5.20.h.

Table 5.17 Parameters of TS and GA for Individual Use

Table 5.17.a TS Setting Values

TS Operator	Setting Value
TS List Size	9
Movement Set	4
Total Iteration	1000
Vector Size	36

Table 5.17.b GA Setting Values

GA Operator	Setting Value
Population size	30
Crossover Probability	0.88
Mutation Probability	0.10
Chromosome Length	36
Total Generation	1200

Table 5.18 Parameters of Hybridization Methods of TS and GA

Model	Operators	Setting Value of model-1	Setting Value of model -2
TS	TS List Size	9	9
	Movement Set	4	N/A
	Total Iteration	1000	N/A
	Vector Size	36	36
GA	Population Size	30	30
	Crossover Probability	0.88	0.88
	Mutation Probability	0.10	0.10
	Chromosome Length	36	36
	Total Generation	1200	1200

Table 5.19 TS and GA Summary Performance

K = 0	
Model	*Total Cost Values (million) SR*
TS	555.550
GA	554.990
TS–GA Model -1	548.070
TS–GA Model -2	538.350

K= 10000			
Model	*Investment Cost SR million*	*Ohmic Losses Cost After the Expansion SR million(*)* *(Calculated for 25 years)*	*Saved Cost By Minimizing Ohmic Power Losses Through the Expansion(**)* *(Calculated for 25 years)*
TS	677.490	582.240	580.16
GA	609.490	644.800	517.56
TS–GA Model -1	655.130	582.720	579.68
TS–GA Model -2	643.630	591.920	570.48

(**) This cost is calculated as the difference between the cost of ohmic power losses before the new line additions (SR million 1,162.40) based on 25 year line's life-time and the cost calculated after the expansion as in (*) for the same period.

Table 5.20.a Best Nominal Configuration Results Using TS at K = 0 with Cost of SR million 555.55

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	13	7 – 13	1
3	14	7 – 16	1
4	16	8 – 22	3
5	17	9 – 11	1
6	22	12 – 14	2
7	23	12 – 23	4
8	24	13 – 18	1
9	25	13 – 20	1
10	35	24 – 25	1
11	36	5 – 25	1

Table 5.20.b Best Nominal Configuration Results Using TS at K=10000 with Cost of SR million 677.490

Addition No	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	4	2 – 3	1
3	8	5 – 17	1
4	14	7 – 16	2
5	16	8 – 22	4
6	21	11 – 14	1
7	22	12 – 14	1
8	23	12 – 23	2
9	24	13 – 18	2
10	25	13 – 20	2
11	26	14 – 22	1
12	28	16 – 18	2
13	29	16 – 20	2
14	34	20 – 21	1
15	35	24 – 25	3
16	36	5 – 25	3

Table 5.20.c Best Nominal Configuration Results Using GA at K = 0 with Cost of (SR) million 554.99

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	14	7 – 16	2
3	16	8 – 22	3
4	22	12 – 14	2
5	23	12 – 23	3
6	24	13 – 18	1
7	32	18 – 23	1
8	35	24 – 25	1
9	36	5 – 25	1

Table 5.20.d Best Nominal Configuration Results Using GA at K=10000 with Cost of SR 609.490

Addition No	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	13	7 – 13	1
3	14	7 – 16	1
4	16	8 – 22	3
5	21	11 – 14	1
6	22	12 – 14	2
7	23	12 – 23	4
8	24	13 – 18	1
9	25	13 – 20	1
10	34	20 – 21	1
11	35	24 – 25	2
12	36	5 – 25	4

Table 5.20.e Best Nominal Configuration Results Using TS – GA (Model – 1) at K = 0 with Cost of SR million 548.07

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	1	1 – 2	1
2	3	1 – 13	1
3	14	7 – 16	2
4	16	8 – 22	3
5	22	12 – 14	2
6	23	12 – 23	3
7	24	13 – 18	1
8	35	24 – 25	1
9	36	5 – 25	1

Table 5.20.f Best Nominal Configuration Results Using TS – GA (Model –1) at K = 10000 with Cost of SR million 655.130

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	4	2 – 3	1
3	8	5 – 17	1
4	14	7 – 16	2
5	16	8 – 22	4
6	21	11 – 14	1
7	22	12 – 14	1
8	23	12 – 23	2
9	24	13 – 18	2
10	25	13 – 20	2
11	28	16 – 18	2
12	29	16 – 20	2
13	34	20 – 21	2
14	35	24 – 25	3
15	36	5 – 25	3

Table 5.20.g Best Nominal Configuration Results Using TS – GA (Model – 2) at K = 0 with Cost of SR million 538.35

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	13	7 – 13	1
3	14	7 – 16	1
4	16	8 – 22	3
5	22	12 – 14	2
6	23	12 – 23	4
7	24	13 – 18	1
8	25	13 – 20	1
9	35	24 – 25	1
10	36	5 – 25	1

Table 5.20.h Best Nominal Configuration Results Using TS – GA (Model – 2) at K = 10000 with Cost of SR million 643.630

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	4	2 – 3	1
3	8	5 – 17	1
4	14	7 – 16	2
5	16	8 – 22	4
6	21	11 – 14	1
7	22	12 – 14	1
8	23	12 – 23	2
9	24	13 – 18	2
10	25	13 – 20	2
11	28	16 – 18	2
12	29	16 – 20	2
13	34	20 – 21	1
14	35	24 – 25	2
15	36	5 – 25	3

5.2.4 ANN Application

The ANN was used to solve the TEP of the 25-bus network. The parameter settings are in Tables 5.21.a, 5.21.b, 5.21.c. The performance of ANN and its hybridization methods are summarized in Table 5.22.

At K = 0, it appears that ANN has a low performance in term of the convergence to the optimal state. The minimum cost that obtained was SR million 758.450. When the ANN is hybridized with the GA, the performance improves. This model reaches to its best values which are SR million 539.09. Moreover, the hybridization of the ANN with TS and GA provided the best minimum costs among all the models which are SR million 535.82. Tables 5.23.a, 5.23.c and 5.23.e show the results of applying the ANN, (ANN and GA) and (ANN, TS and GA) respectively.

On the other hand, at K=10000, the cost of ohmic power losses was also considered in the calculation to obtain the new network configuration that minimizes these losses. By using ANN, the investment cost is SR million 840.670 and the saved cost SR million 554.92. The results are improved when the ANN and GA are hybridized. The investment cost is SR million 650.070 and the cost of the ohmic losses after the expansion is SR million 582.180 which means it can save SR million 580.220 for 25 years compared to the losses before the additions. Finally, the hybridization of ANN, TS and GA obtains the best results among all the methods. The investment cost is SR million 632.940 and it minimizes the cost of ohmic losses to SR million 571.200. Tables 5.23.b, 5.23.d and

5.23.f show the results of applying the ANN, (ANN & GA) and (ANN, GA & TS) respectively.

Figures 5.12 shows that only 15 lines are needed to solve the TEP of the system in the case of excluding the power loss cost while Figures 5.13 show that 30 lines are required to solve TEP with the power loss consideration using this best hybridization method.

Table 5.21.a ANN Setting Values

ANN Parameter	Setting Value
Neurons of the Input Layer	1
Neurons of the Hidden Layer	3
Neurons of the Output Layer	36
The Training Set at Each Iteration	50
Training Error	0.08
Total Number of Iteration	2000

Table 5.21.b ANN and GA Setting Values

Model	The Parameter	Setting Value
ANN	Neurons of the Input Layer	1
	Neurons of the Hidden Layer	3
	Neurons of the Output Layer	36
	The Training Set at Each Iteration	50
	Training Error	0.08
	Total Number of Iteration	2000
GA	Population Size	30
	Crossover Probability	0.88
	Mutation Probability	0.10
	Chromosome Length	36
	Total Generation	1200

Table 5.21.c (ANN, TS and GA) Setting Values

Model	The Parameter	Setting Value
ANN	Neurons of the Input Layer	1
	Neurons of the Hidden Layer	3
	Neurons of the Output Layer	36
	The Training Set at Each Iteration	50
	Training Error	0.08
	Total Number of Iteration	2000
TS	TS List Size	9
	Vector Size	36
GA	Population Size	30
	Crossover Probability	0.88
	Mutation Probability	0.10
	Chromosome Length	36
	Total Generation	1200

Table 5.22 Summary Performance of ANN and its Hybridization Methods

K = 0	
Model	***Total Cost SR million***
ANN	758.540
ANN with GA	539.030
ANN With GA &TS	535.820

K =10000			
Model	*Investment cost SR million*	*Ohmic Losses cost after the expansion SR million(*)* *(Calculated for 25 years)*	*Saved cost by minimizing ohmic power losses through the expansion(**)* *(Calculated for 25 years)*
ANN	840.670	607.480	554.920
ANN with GA	650.070	582.180	580.220
ANN , GA and TS	632.940	571.200	591.200

(**) This cost is calculated as the difference between the cost of ohmic power losses before the new lines additions (SR million 1,162.40) based on 25 years line's life-time and the cost calculated after the expansion as in (*) for the same period.

Table 5.23.a Best Nominal Configuration Results Using ANN at K = 0 with Cost of SR million 758.54

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	1	1 – 2	1
2	3	1 – 13	1
3	8	5 – 17	2
4	14	7 – 16	2
5	16	8 – 22	4
6	17	9 – 11	1
7	18	9 – 15	1
8	21	11 – 14	1
9	22	12 – 14	1
10	23	12 – 23	1
11	24	13 – 18	2
12	25	13 – 20	1
13	27	15 – 22	1
14	28	16 – 18	2
15	29	16 – 20	1
16	32	18 – 23	2
17	34	20 – 21	2
18	35	24 – 25	3
19	36	5 – 25	1

Table 5.23.b Best Nominal Configuration Results Using ANN at K = 10000 with Cost of (SR million) 840.670

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	1	1 – 2	1
2	3	1 – 13	1
3	4	2 – 3	1
4	8	5 – 17	2
5	14	7 – 16	2
6	15	8 – 16	1
7	16	8 – 22	4
8	17	9 – 11	1
9	18	9 – 15	1
10	21	11 – 14	1
11	22	12 – 14	1
12	23	12 – 23	1
13	24	13 – 18	2
14	25	13 – 20	1
15	27	15 – 22	1
16	28	16 – 18	2
17	29	16 – 20	1
18	32	18 – 23	2
19	34	20 – 21	2
20	35	24 – 25	3
21	36	5 – 25	1

Table 5.23.c Best Nominal Configuration Results Using ANN – GA at K = 0 with Cost of SR million 539.030

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	14	7 – 16	2
3	16	8 – 22	3
4	22	12 – 14	2
5	23	12 – 23	3
6	24	13 – 18	2
7	29	16 – 20	1
8	35	24 – 25	1
9	36	5 – 25	1

Table 5.23.d Best Nominal Configuration Results Using ANN – GA at K = 10000 with Cost of SR million 650.070

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	4	2 – 3	1
3	8	5 – 17	1
4	14	7 – 16	2
5	16	8 – 22	4
6	21	11 – 14	1
7	22	12 – 14	1
8	23	12 – 23	2
9	24	13 – 18	2
10	25	13 – 20	2
11	28	16 – 18	2
12	29	16 – 20	2
13	34	20 – 21	1
14	35	24 – 25	2
15	36	5 – 25	4

Table 5.23.e Best Nominal Configuration Results Using ANN with (GA and TS) at K = 0 with Cost of SR million 535.820

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	13	7 – 13	1
3	14	7 – 16	1
4	16	8 – 22	3
5	22	12 – 14	2
6	23	12 – 23	3
7	24	13 – 18	1
8	25	13 – 20	1
9	35	24 – 25	1
10	36	5 – 25	1

Table 5.23.e Best Nominal Configuration Results Using ANN with (GA and TS) at K =10000 with Cost of SR million 632.940

Addition No.	Rights-of-Way	Bus Link	No. of Extra Added Lines
1	3	1 – 13	1
2	14	7 – 16	1
3	16	8 – 22	3
4	21	11 – 14	1
5	22	12 – 14	2
6	23	12 – 23	3
7	24	13 – 18	4
8	25	13 – 20	3
9	28	16 – 18	2
10	29	16 – 20	3
11	34	20 – 21	1
12	35	24 – 25	2
13	36	5 – 25	4

5.2.5 Summary performance

Table 5.24 shows the summary performance of all the methods. It appears that the combination of ANN, GA and TS has the best performance in term of obtaining the optimal solution for the two cases (K = 0 and K = 10000) while ANN is the worst among all the models. LP and QP did not produce the optimal solution because it might stick with the local minimal value. Although the starting conditions for them have been changed, little improvement is observed. However, GA has converged to a better solution when compared to TS but the hybridization between both algorithms leads improving the results. Also, the performance of the hybridization model between GA and ANN does not lead to the optimal solution but its results are better than the results obtained from these algorithms when they were applied individually.

Table 5.24 Summary Performance of AI Applied to 25-Bus IEEE

K = 0	
Model	*Best Cost Value (SR) million*
LP	638.250
TS	555.550
GA	554.990
TS& GA (1)	548.070
TS &GA (2)	538.350
ANN	758.540
ANN & GA	543.420
ANN, GA &TS	**538.350**

K =10000			
Model	*Investment cost SR million*	*Ohmic Losses cost after the expansion SR million(*)* (Calculated for 25 years)	*Saved cost by minimizing ohmic power losses through the expansion(**)* (Calculated for 25 years)
QP	680.470	741.590	420.800
TS	677.490	582.240	580.160
GA	609.490	644.800	517.560
TS& GA (1)	655.130	582.720	579.680
TS &GA (2)	643.630	591.920	570.480
ANN	840.670	607.480	554.920
ANN & GA	630.070	582.180	580.220
ANN, GA &TS	**632.940**	**571.200**	**591.200**

(**) This cost is calculated as the difference between the cost of ohmic power losses before the expansion of the transmission system in the network (SR million 1,162.40) based on 25 year line's life-time and the cost calculated after the expansion as in (*) for the same period.

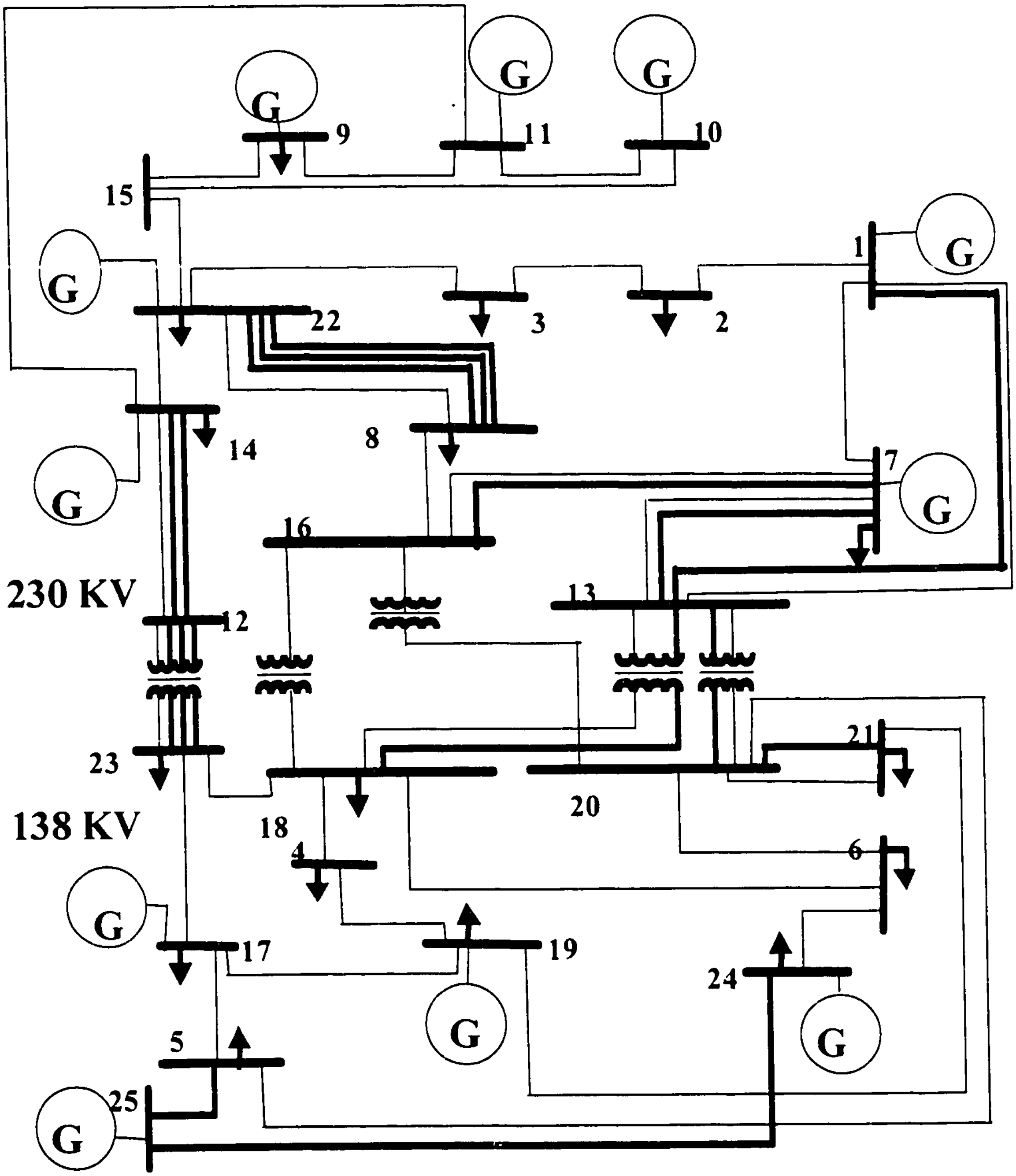


Figure 5.12 Best Configuration results When a Combination of ANN, GA and TS Was Used at K=0

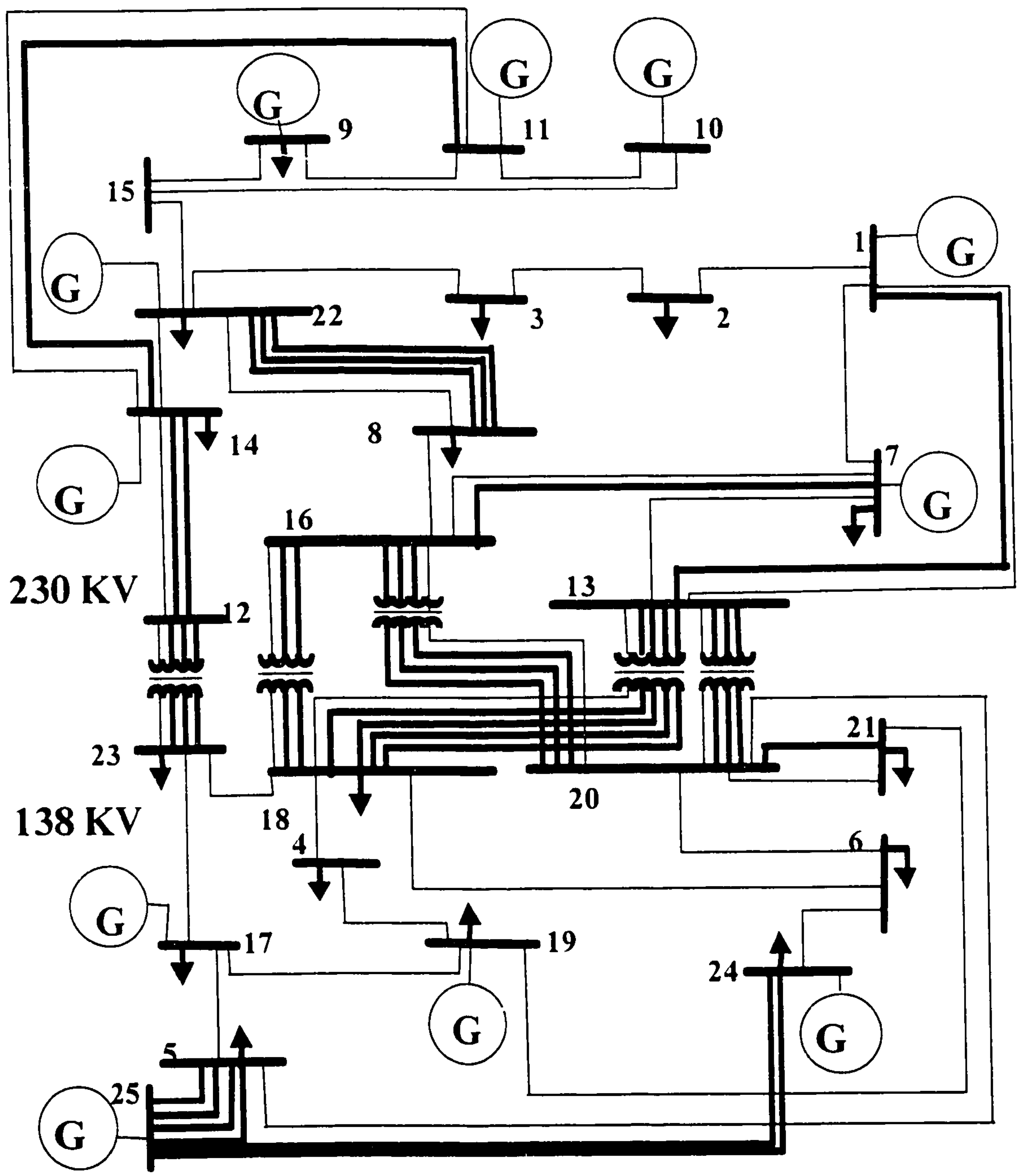


Figure 5.13 Best Configuration Results when a Combination of ANN, GA and TS Was Used at K=10000

5.3 **Real Power System Application**

This section presents the results of the work undertaken to address the TEP for one portion of SCECO-EAST network [43]. The study deals with the expansion from the year 1999 to the year 2010. The 380KV-transmission line portion of the SCECO-EAST network is going to be studied through the application of hybridization method between TS, GA and ANN.

5.3.1 System Description

This system is going to be studied under the following assumptions:

1. Reactance of the new circuits is going to be chosen from Table 5.27 and a *maximum of 4 extra lines are* allowed in each of *the 14 possible rights-of-way with equal importance of all transmission lines* ($n_{ij}^{max} = 4$).
2. The maximum phase angle and the maximum difference of phase angle between the buses are assumed *as 20 degrees*. ($|\Delta\theta^{max}| = |\theta^{max}| = 20$ degrees)
3. The maximum power flow limit ($P_i^{max} = 1250$ MW) and the reactance (X_{ij}) with the cost (c_{ij}) of the new additional lines are listed in table 5.27.
4. Power losses have been calculated in the objective function for four cases, when the cost of the kWh is assumed to be SR 0.050, 0.10, 0.15 and 0.20. Thus the loss coefficients K are approximately 10000, 20000, 30000 and 40000 respectively. The p.u base in the DC-Load Flow analysis is 100 MVA and the base cost of the added lines is SR 10,000.

The existing initial plan, as developed by SCECO-EAST, is shown in Figure 5.14 [43]. Table 5.25 shows the line data while table 5.26 provides the load and generation in the year 2010. As per the utility plan, it is clear that there will be three buses to be added to the network configuration for year 1999. These are Ras Az Zawr Power Plant, Qaysumah and King Khaled Military City. The bus information is explained in highlighted rows in tables 5.25 and 5.26.

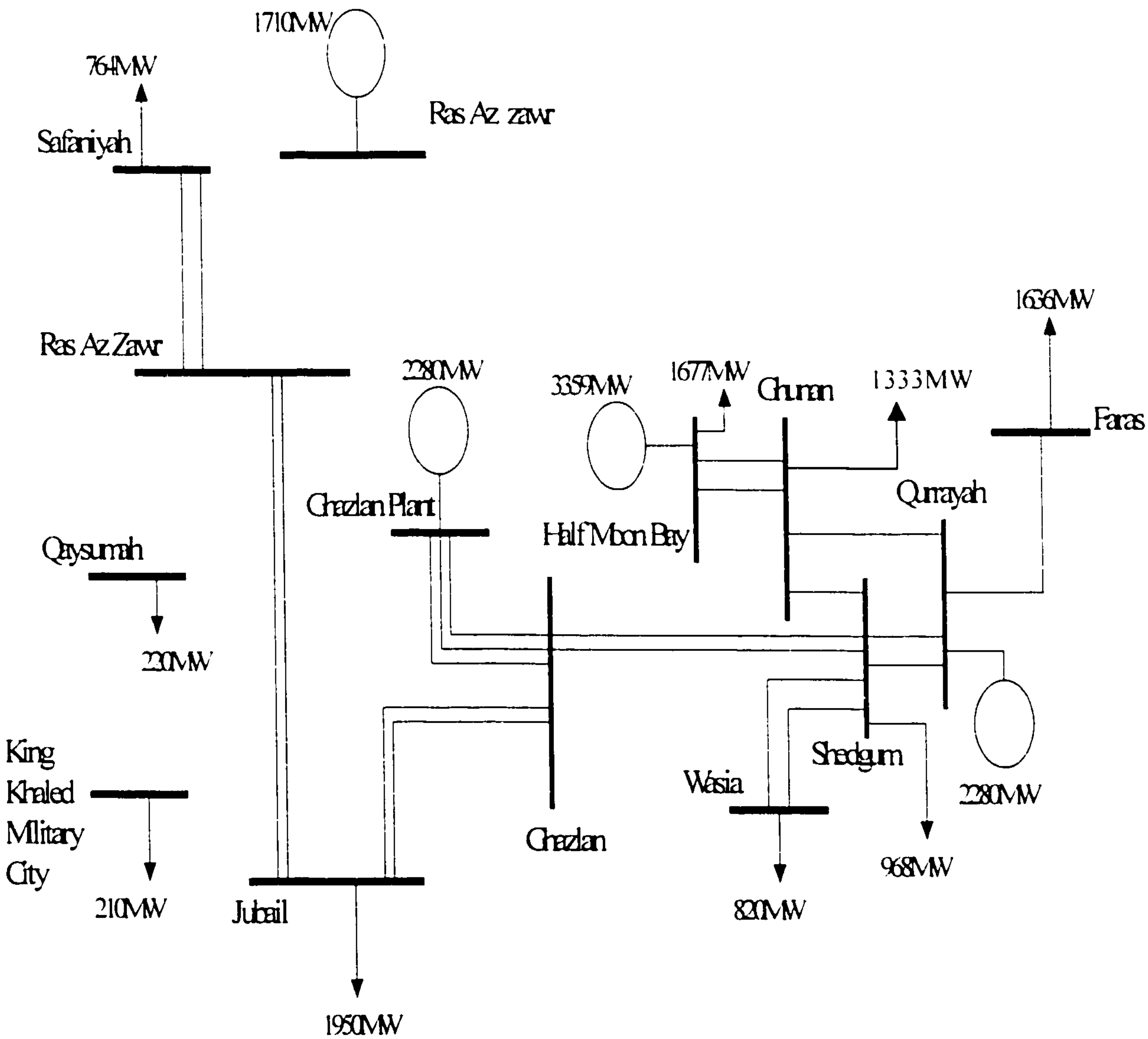


Figure 5.14 Initial Network Plan

Table 5.25 380KV line data

Right of Way	Bus Link	Distance (KM)	Existing Circuits	Resistance (Ω)	Reactance (Ω)	Cost X 10000 (SR)
1	Faras – Qurrayah	136	1	0.0072	0.02924	29349
2	Qurrayah – Shedgum	84	2	0.0045	0.01806	18127
3	Qurrayah – Ghunan	43	1	0.0023	0.00925	9279
4	Shedgum – Wasia	240	2	0.0129	0.05160	51792
5	Shedgum – Ghunan	74	1	0.0040	0.01591	15969
6	Shedgum – Ghazlan	180	2	0.0097	0.03870	38844
7	Ghunan – Half Moon Bay	40	2	0.0022	0.00860	8632
8	Ghazlan – Ghazlan Plant	50	3	0.0027	0.01075	10970
9	Ghazlan – Jubial	50	2	0.0027	0.01075	10790
10	Jubial – Ras Az Zawr	90	2	0.0048	0.01935	19422
11	Ras Az Zawr – Safaniyah	50	2	0.0027	0.01075	10790
12	***Ras Az Zawr – Ras Az Zawr Plant***	***60***	***1***	***0.0034***	***0.01356***	***12948***
13	***Ras Az Zawr – Qaysumah***	***335***	***2***	***0.0019***	***0.07571***	***72293***
14	***Qaysumah – King Khaled Military City***	***80***	***2***	***0.0045***	***0.01808***	***17264***

Table 5.26 380KV Load/Generation data

Bus Name	Bus Number	Generation (MW)	Load (MW)
Faras	1	-	1636
Qurrayah	2	2280	-
Shedgum	3	-	968
Wasia	4	-	820
Ghunan	5	-	1383
Half Moon Bay	6	1681	-
Ghazlan	7	-	-
Ghazlan Plant	8	2280	-
Jubial	9	-	1950
Ras Az Zawr	10	-	-
Safaniyah	11	-	764
Ras Az Zawr Plant	***12***	***1710***	-
Qaysumah	***13***	-	***220***
King Khaled Military City	***14***	-	***210***

5.3.2 Discussion of Utility's Result

The concern in this TEP is how to handle the expected load increase in the network in 2010. As a result, three buses are proposed to be added, one bus which is a new generation in Ras Az Zawr while the other two are new load demand locations. The new generation location was selected because it is the nearest place to the load in the Jubail Bus and it is the nearest location to the Gulf which is required for the steam turbine. Also, the link between Ras Az Zawr and Qaysumah was established due to the closest bus connection. This link is 335KM long while the other alternative, that between Qaysumah and Jubail, is 400KM long. The link between Qaysumah and King Kahled Military City is chosen because of geographical reasons.

The TEP of the network was solved by the utility and the development plan is shown in Figure 5.15. The cost that was proposed by SCECO-EAST to fit this expansion is SR 3,392,380,000. The details of the solution are shown in Table 5.27. The author does not have access the methodology and the algorithm used by the utility.

Table 5.27 SCECO-EAST Line Addition Locations.

Addition No.	*Bus Link*	*No. of additional lines using the utility method*
1	Faras – Qurrayah	2
2	Jubail - Ghazlan	1
3	Ras Az Zawr – Jubial	2
4	Ras Az Zawr - Ras Az Zawr Plant	4
5	Ras Az Zawr - Qaysumah	2
6	Qaysumah – King Khaled Military City	2

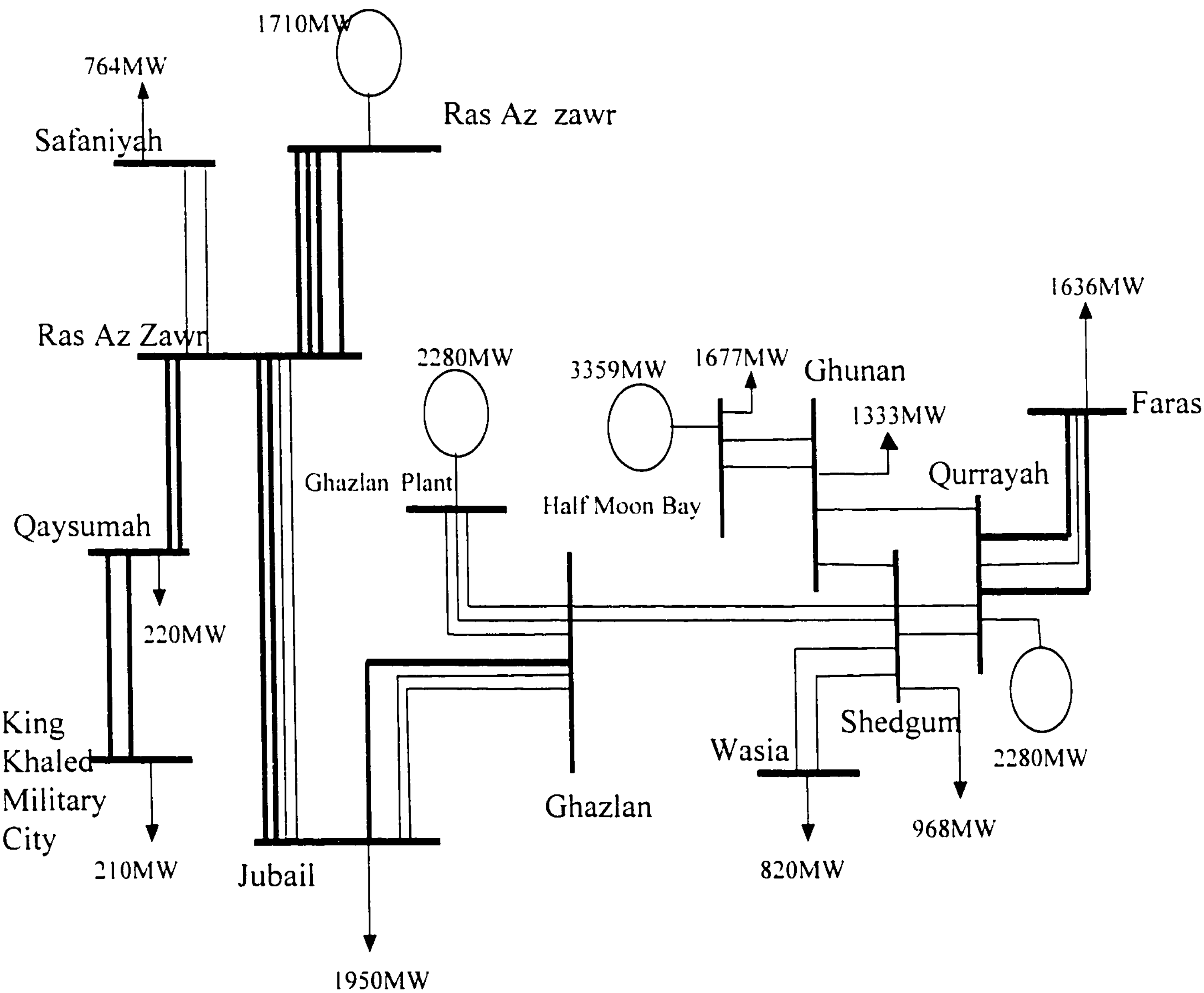


Figure 5.15 SCECO-EAST Solution

5.3.3 Proposed Algorithm Application

The proposed hybrid method that combines the ANN, TS and GA was used to solve the TEP of the network. The parameter settings are in Table 5.28.The results of applying the method are summarized in Table 5.29.

If power cost losses are excluded in the planning calculations, the total cost reaches to SR million 1448.02 with the new additions that are shown in Figure 5.16.

On the other hand, this method was applied for four cases when the power losses are included. In the case of assuming the cost of kWh to be SR 0.05, K=10000, the configuration of the new network is SR million 2,065.21. These additions will minimize the cost of the ohmic power losses from SR million 5,458.60 to SR million 3290.40. At K= 20000, the investment cost reaches to SR million 2,790.30. This cost leads to save SR million 5,265.60 by minimizing the power loss cost over 25 years compared to the losses before the expansion. When one kWh is sold by SR 0.15 the network will have losses that cost SR million 16,376.00 over 25 years. As a result of the expansion, the investment cost of the new lines is SR million 3,221.90 and the cost of the losses over 25 years will be minimized to SR million 794,5.40. Finally, at K = 40000, the investment cost reaches to SR million 5,360.45. Knowing that the cost of the ohmic power losses before the addition is SR million 21,835.00, the new scenario leads to saving SR million 13,704.48 compared to the cost of the ohmic power losses after the expansion which is SR 813,0.20. Figures 5.17, 5.18, 5.19 and 5.20 show the network configuration for each case while Table 5.29 shows the results in detail.

Table 5.28 Proposed Method Parameters

Model	The Parameter	Setting Value
ANN	*Neurons of the Input Layer*	1
	Neurons of the Hidden Layer	3
	Neurons of the Output Layer	14
	The Training Set at Each Iteration	50
	Training Error	0.08
	Total Number of Iteration	1000
TS	TS List Size	9
	Vector Size	14
GA	Population Size	20
	Crossover Probability	0.82
	Mutation Probability	0.09
	Chromosome Length	14
	Total Generation	1000

Table 5.29 Summary Performance of ANN, GA and TS

Cost of SR/kWh	*Corresponding K*	*Investment Cost SR million*
Not-considered	0	1,448.02
0.05	10000	2,065.21
0.10	20000	2,790.30
0.15	30000	3,221.90
0.20	40000	5,360.45

Cost of SR/kWh	*Ohmic Losses Cost before The New Lines Additions SR million(*1) (Calculated for 25 years*	*Ohmic Losses Cost AfterThe Expansion SR million(*2) (Calculated for 25 years)*	*Saved Cost By Minimizing Ohmic Power Losses Through The Expansion(**) (Calculated for 25 years)*
0.05	5,458.60	3,290.40	2,166.20
0.10	10,917.00	5,651.40	5,265.60
0.15	16,376.00	7,945.40	8,430.60
0.20	21,835.00	8,130.20	13,704.80

*(**) This cost is calculated as the difference between the cost of ohmic power losses before the new lines additions (*1) and the cost calculated after the expansion as in (*2).*

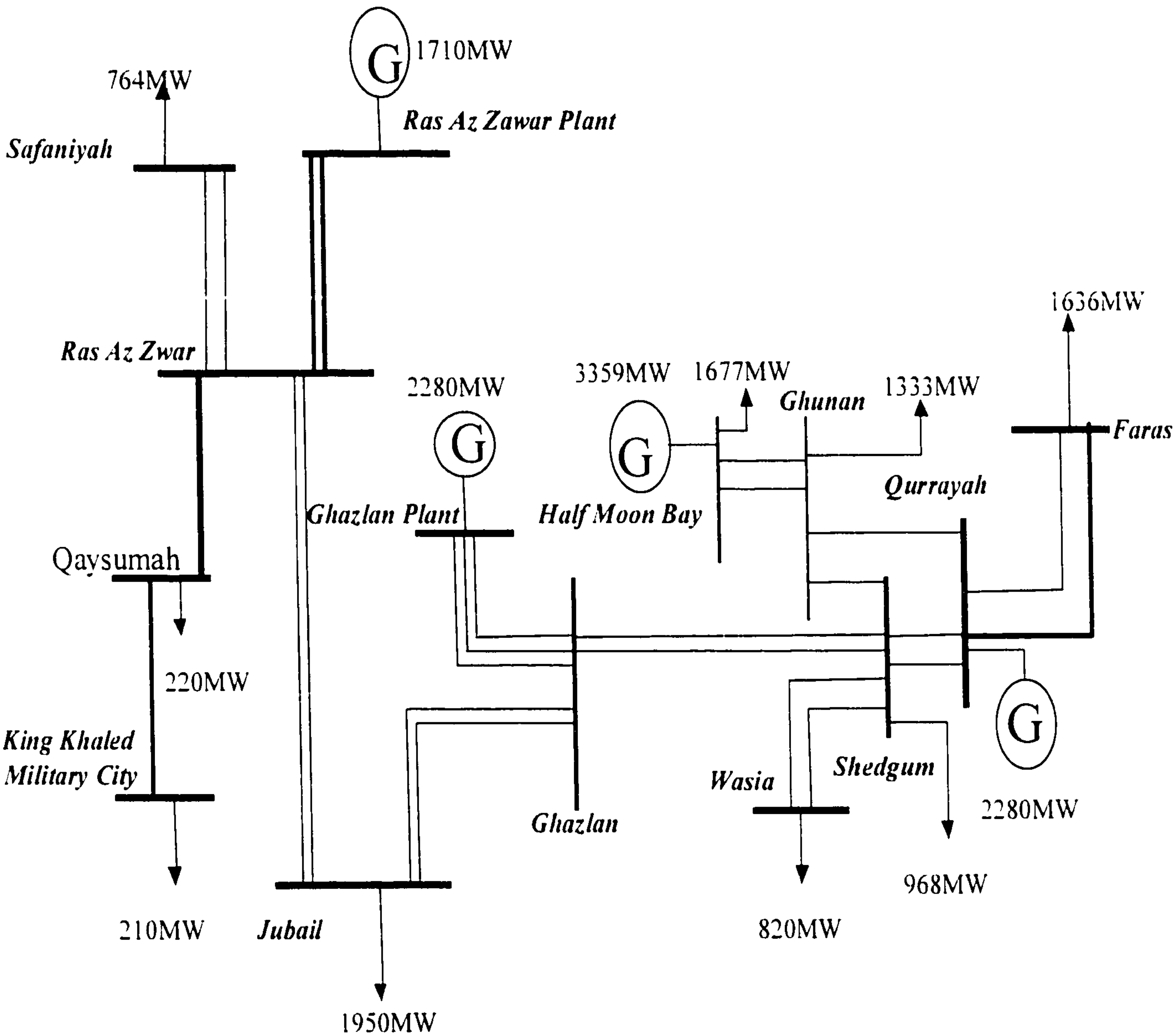


Figure 5.16 Final Proposed Solution at K=0

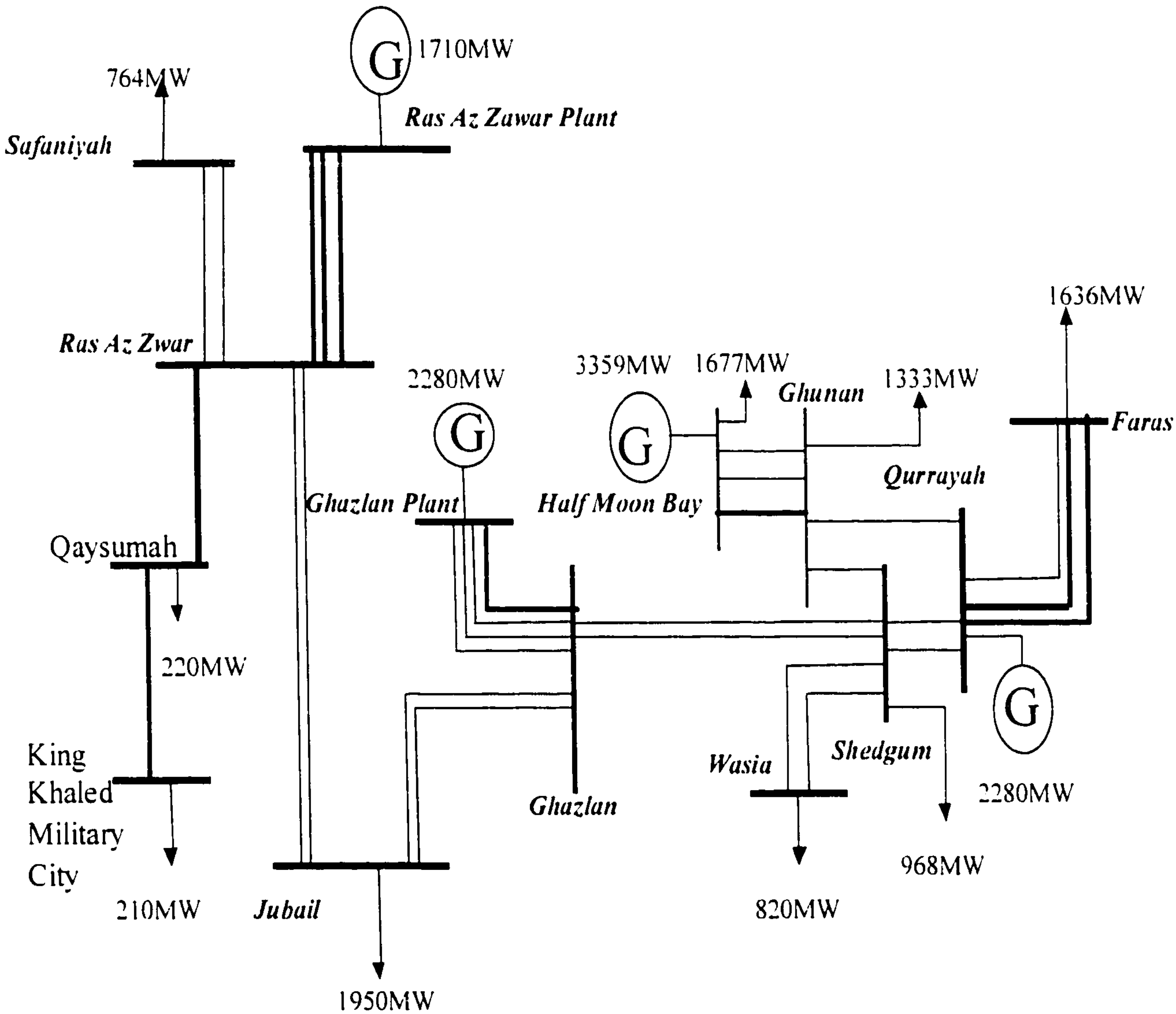


Figure 5.17 Final Proposed Solution at K=10000

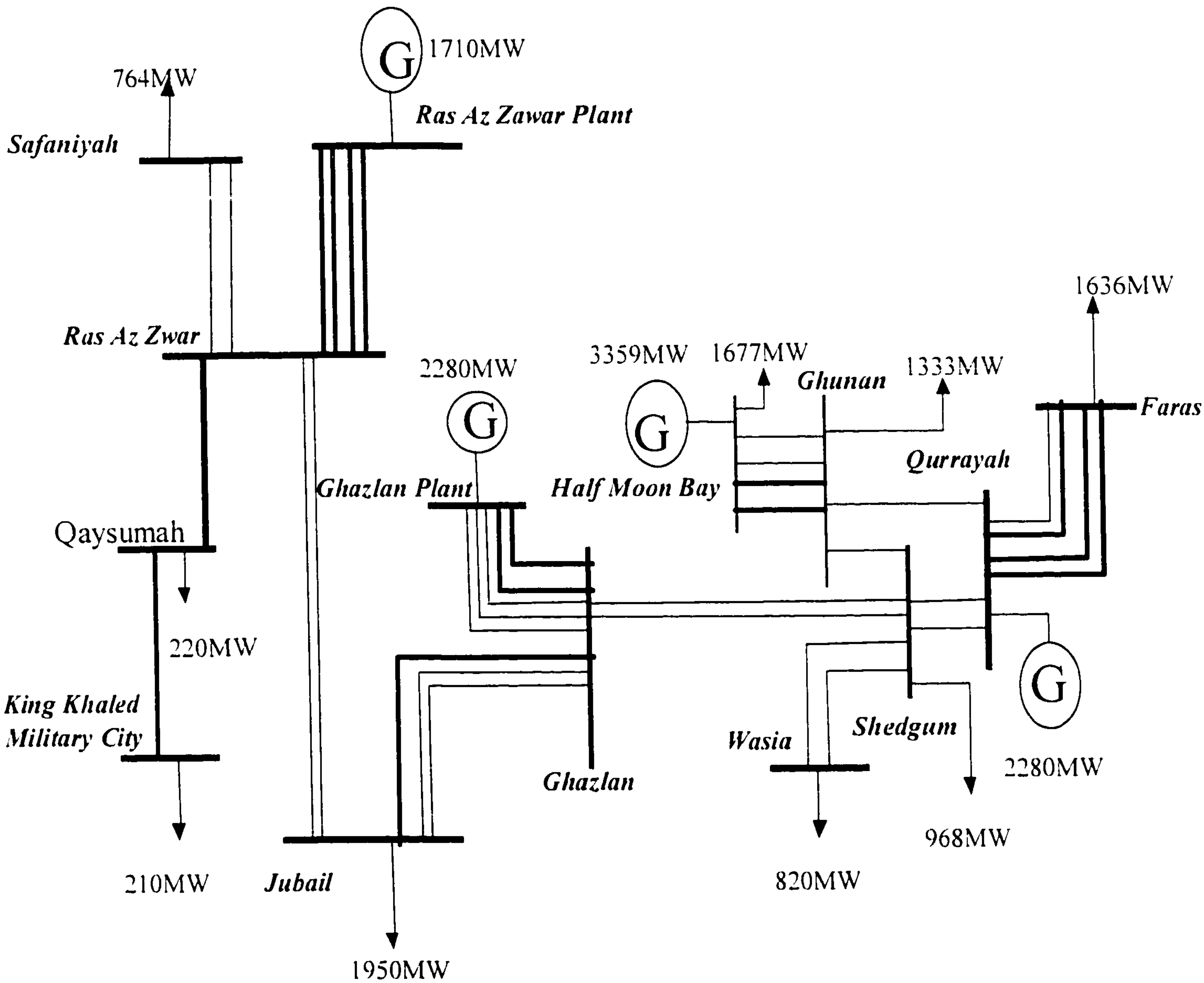


Figure 5.18 Final Proposed Solution at K= 20000

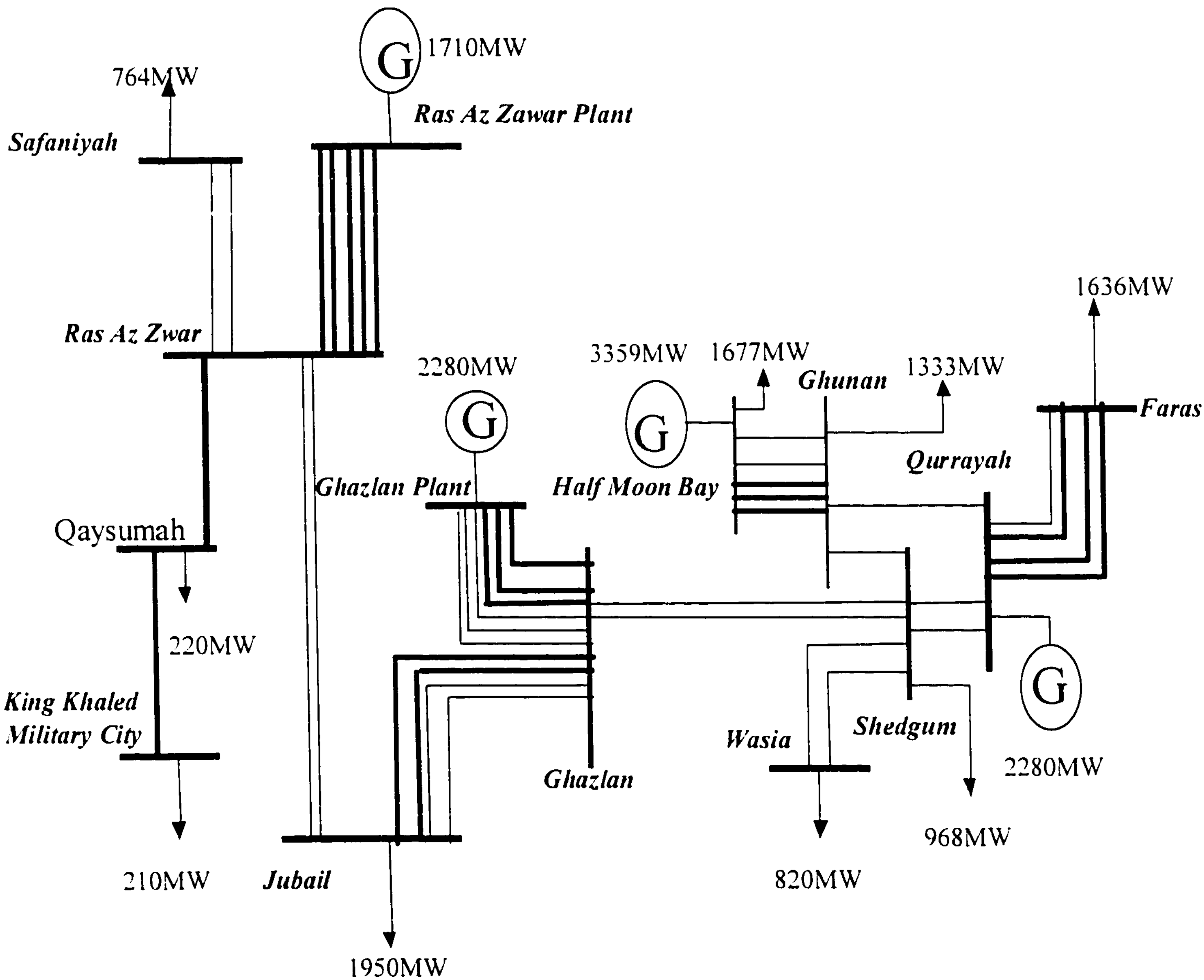


Figure 5.19 Final Proposed Solution at K= 30000

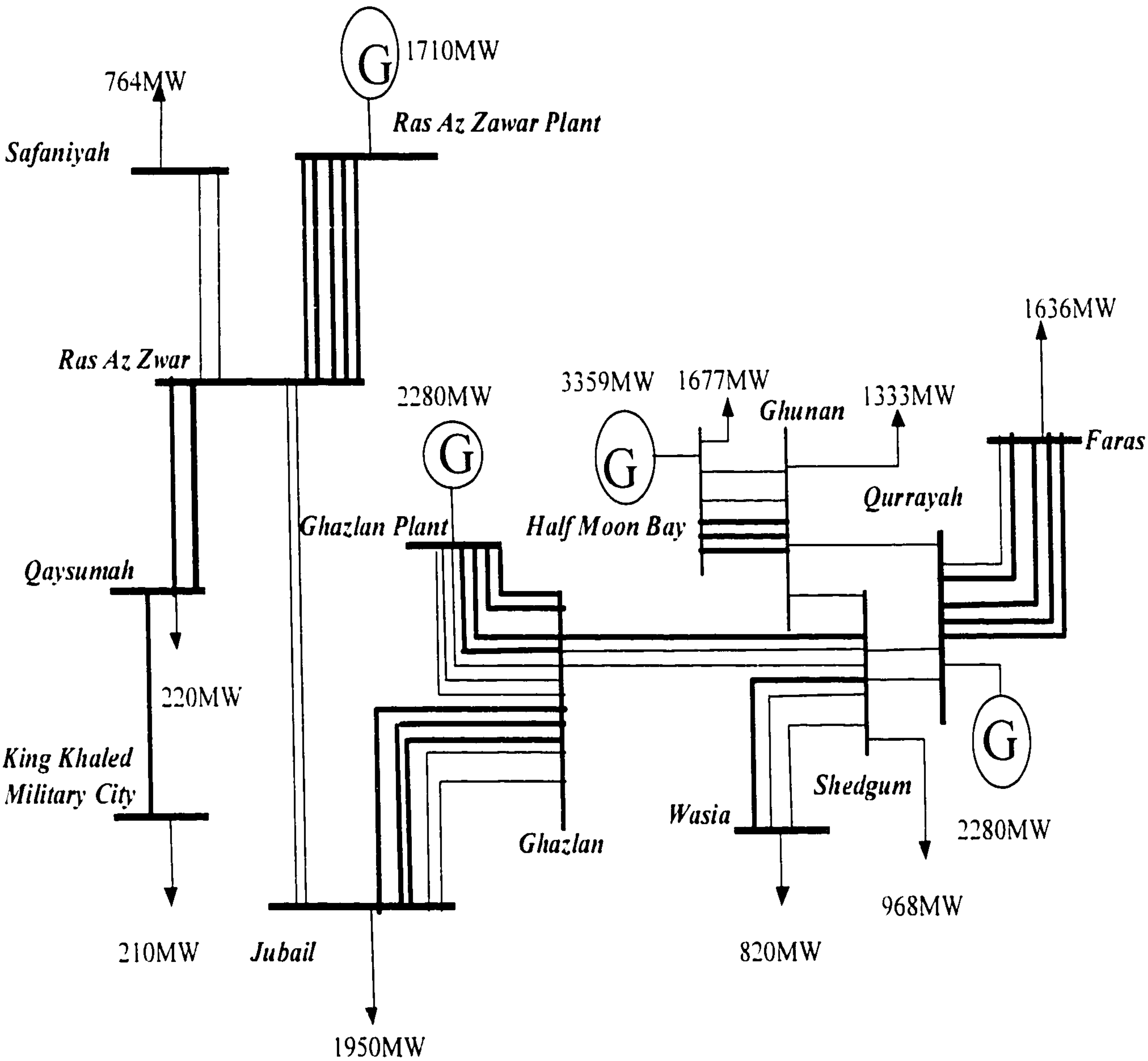


Figure 5.20 Final Proposed Solution at K= 40000

CHAPTER - 6

CONCULSION AND FUTURE WORK

6.1 *Concluding Remarks*

The power system Transmission Expansion Problem (TEP) had been formulated as an optimization problem. The objective was to minimize the transmission investment costs that handle the increased load and the additional generation requirements in term of line additions and ohmic power losses. Several constraints were considered including the power flow on the network lines, the Right-of-Way's validity and its maximum line addition, and maximum angle change across the buses. The TEP was then solved using Artificial Intelligent (AI) tools like the Genetic Algorithm, Tabu Search and Artificial Neural Network against Linear and Quadratic Programming Models. Based on the quality of the final solution and computational speed, these methods have proved to be suitable for solving difficult optimization problem.

The effectiveness of the AI methods in dealing with small and large-scale systems is tested through the applications of six-bus system, the IEEE-25 bus network and the SCECO-EAST network. For the 6-bus system, all the proposed models have been applied and the optimum design was obtained compared to the linear and quadratic programming approaches. For the IEEE – 25 Bus system, the hybridization of TS, GA and ANN proved to be more effective than other approches.

The TEP of SCECO – EAST network from 1999 to 2010 was also solved using the best-tested method (hybridization of all AI). It provided a future network configuration for four different cases.

To address the thesis issues, the hybridization of GA, TS and ANN has the following features:

1. Its results confirm that it is superior in dealing with a large-scale problem in which the size of search spaces increases exponentially with the dimension of the network.
2. It considers the investment cost in terms of line additions and ohmic power losses to be minimal under technical and economical constraints.

6.2 Recommendations for Further Research

It is recommended to add some terms in the objective function, like the financial risk rate and the importance of the expanded links, to modify the proposed method usage with the extended problem. This will help to prove that the proposed method can handle more constraints in less time.

Additional area of improvement would be to study and to formulate the problem of selection of the best location of new added buses that minimize the new added lines.

Since the Fuzzy Logic (FL) is not yet applied in the TEP, the idea of using FL to solve the TEP can enhance the study. It might improve the performance of the proposed algorithm in terms of the ability to converge to an optimum solution or in terms of convergence speed.

REFERENCES

1. Turan Gonen. Electric Power Transmission System Engineering Analysis and Design. *John Willy and Sons, Inc., USA. 1988.*

2. Marija Ilic, Francisco Galiana, Lester Fink, Anjan Bose, Pierre Mallet and Hisham Othman." Transmission Capacity in Power Network" *Electrical Power and Energy Systems*, Vol.20, No.2, pp-99-110, 1998

3. Transmission System, Electrical Power System-I, *Saudi ARAMCO Engineering Encyclopedia*, pp.47-60

4. R. Billinton and J. Oteng-Adjei, " Utilization of Interrupted Energy Assessment Rates in Generation and Transmission System Planning " , *IEEE Tansaction on Power Systems*, Vol. 6, No.3 , PP. 1245-1253, 1991.

5. M. E. El-Hawary and G.S. Christensen, Optimal Operation of Electric Power System, *Academic Press, New York, USA. 1979.*

6. W. L. Weeks, Transmission and Distribution of Electrical Energy, *Harper and Row Publishers, New York, USA.*

7. V. A. Levi and M. S. Calavic, "Linear- Programming Based Decomposition Method for Optimal Planning of Transmission Network Investment" *IEE Proceeding-C*, Vol. 140, No. 6, pp.516-522, November 1993.

8. Kern J. Kim, Young M. Park and Kuang Y. Lee," Optimal Long Term Transmission Expansion Planning Based On Maximum Principle", *IEEE Transaction on Power Systems*, Vol. 3 , No.4,November 1988.

9. R. Villosana, L. L. Garver and S. J. Solon. " Transmission Network Planning Using Linear Programming" *IEEE Ttransaction on Power Apparatus and System*, Vol. PAS-104, No. 2, PP. 349- 356, February 1985.

10. Sharifnia and H. Z. Aa shtiani." Transmission Network Planning : A Method for Synthsis of Minimum-Cost Secure Networks". *IEEE Transaction on Power Apparatus and Systems*, Vol . PAS-104, No. 8, PP.2026-2034, 1985.

11. Jin-Cherng Lin and Chyan-Goei Chang,"Zero-One Integer Programming Model in Path Selection Problem of Structural Testing" , *IEEE Transaction on computer software and application* , September 1989,PP. 618-627.

12. R. Romero and A. Monticelli, " A Zero- One Implicit Enumeration Method for Optimizing Investments in Transmission Expansion Planning", *IEEE Transactions on Power Systems*, Vol. 9, No.3, PP. 1385-1391, August 1994.

13. Michel L. GiMes, "Optimum HVDC-Transmission Expansion Planning- A New Formulation" *IEEE- Transaction on Power System*. Vol.PWRS-2, No.2, May 1987, PP.429-435

14. Z. M. Al-Hamouz, "Transmission Network Planning Using Quadratic Programming" Master's Thesis, Jordan University of Sciences and Technology, 1989

15. M. V. Pereira, L. M. Pinto, S. H. Cunha and G. C. Aliveira, "A Decomposition Approach to Automated Generation/ Transmission Expansion Planning" *IEEE-Transaction on Power Apparatus and Systems*, Vol.PAS-104, 11 November 1985, PP. 3074-3081.

16. Viktor A. Levi and Milan S. Calovic, " A New Decomposition Based Method for Optimal Expansion Planning of Large Transmission Networks", *IEEE Transaction on Power System* , Vol. 6, No. 3, August 1991.

17. R. Romero and A. Monticelli, " A Hierarchical Decomposition Approach for Transmission Network Expansion Planning" *IEEE Transaction on Power Systems* , Vol 9 , No. 1 , PP. 373-380, February 1994.

18. Yuan-Yih Hsu, Pei- Hwa Huang, Chia-Jen Lin and Chiang-Tsung Huang,"Oscillatory Stability Considerations in Transmission Expansion Planning", *IEEE Transaction on Power System* , Vol. 4, No. 3,PP. 1110-1114, August 1989.

19. Mahmoud M. El-Metwally and Ahmad M. Harb"Transmission Planning Using Admittance Approach and Quadratic Programming", *Electrical Machines and Power Systems*, Vol.21, PP. 61-83, 1993.

20. Gerardo Latorre and Ignacio J. Perez, "CHOPIN, A Heuristic Model for Long Term Transmission Expansion Planning" *IEEE Transaction on Power Systems*, Vol. 9, No.4, PP.1886-1894, November 1994.

21. R. Romero, R. A. Gallego and A. Monticelli, " Transmission System Expansion Planning By simulated Anealing", *IEEE Transaction on Power System*, Vol. 11, No. 1, pp. 364-369, February 1996.

22. R. Romero, R. A. Gallego and A. Monticelli, "Parallel Simulated Annealing Applied to Long Term Transmission Network Expansion Planning" *IEEE Transaction on Power Systems*, Vol. 12 , No.1, Feb. 1997.

23. Hugh Radnick, Rodrigo Palam, Eliana Cura and Carlas Silva, "Economically Adapted Transmission Systems in Open Acess Schemes- Application of Genetic Algorithms" *IEEE Transaction on Power Systems*, Vol. 11, No.3, PP. 1427- 1440, August 1996.

24. R. Romero, R. A. Gallego and A. Monticelli, "Transmission System Expansion Planning by an Extended Genetic Algorithm" *IEEE Transaction on Power Systems*, Vol. 145, No.3, pp. 329-335, May. 1998.

25. Katsuhisa Yoshimoto, Keiichiro Yasuda and Ryuichi Yokoyama, "Transmission Expansion Planning Using Neuro-Computing Hypridized with Genetic Algorithm", *IEEE Transaction on Neural Network*, Vol. 1, December 1995, PP. 126-131.

26. R. Romero, R. A. Gallego and A. Monticelli, " Comparative Studies on Non-Convex Optimization Methods for Transmission Network Expansion Planning" *IEEE Transaction on Power Systems*, Vol. 13, No.3, pp.822-828,August 1998.

27. Baldick, R. and Kahn, E. "Transmission Planning Issues in a Competitive Economic Environment" *IEEE Transactions on Power System*, Vol.8, No.4, November 1993, pp.1497-1503.

28. Vojdani, A., Imparato, C., Saini, N., Wollenberg, B. and Happ, H. "Transmission Access Issues." *IEEE Transaction on Power System*, Vol.11, No.1, February 1996, pp. 41-51.

29. Hughes, W and Felak, R. " Bridging the Gap between Theory and Practice of Transmission Pricing" Electricity Transmission Pricing and Techology, Einhorn, M and Siddiqi, R., eds.., Kiuwer Academic Publishers, Boston, pp.25-58, 1996.

30. Marangon Lima, J., Pereira, M. and Pereira, J. L." An Integrated Framework for Cost Allocation in a Multi-owned Transmission System" IEEE Transaction On Power System, Vol. 10, No.2, pp.971-977, May 1995.

31. Tsukamoto, Y., Iyoda I. "Allocation of Fixed Transmission Cost to Wheeling Transactions by Cooperative Game Theory" *IEEE Transaction on Power System*, Vol.11, No.2, pp.620-629, May 1996

32. D. E. Goldberg, Genetic Algorithm in Search, Optimization and Machine Learning, Reading, Mass., *Addison Wesely, 1989.*

33. Zbigniew Michalewicz. Genetic Algorithm + Data Structures = Evolution Programs, *Springer-Verlag Berlin Heidelberg , New York , USA, 1992.*

34. L. Davis, Handbook of Genetic Algorithms, *Van Nostrand, New York , 1991*

35. Richard L. Daniels and Joseph B. Mazzola, "A Tabu Search Heuristic for the Flexible-Resource Flow Shop Scheduling Program", *Annals of Operation and Research.* 41(1993), 207-230

36. Mauro Dell Amico and Marco Trubian,"Applying Tabu Search to the Jop-shop Scheduling Problem" *Annals of Operation and Reserch* . 41(1993), 253-278.

37. M. Laguna, T. Feo and H. Elrod,"Genentic Algorithm Ang Tabu Search: Hybrids for Optimization" *Operations Research*, Vol. 42, no. 4, pp. 677-687 1994.

38. T. T. Al-Saba and I. M. El-Amin, "Artificial Neural Network as Applied to Long-term Demand Forecasting" *Artificial Intelligence in Engineering, Vol.13, Issue. 2, 27-January – 1999, PP. 189-197.*

39. Parlos, *A. G.;* Oufi, E ; Muthusami, J ; Patton, A and Atiya, A "Development of an Intelligent Long-term Electrical Load Forecasting System" *Intelligent system applications to power system proceeding, 1996, PP.288-292.*

40. Highlay, D. and Hilmns, T. " Load Forecasting by ANN" *Computer Application in Power, IEEE*, Vol. 6, No. 3, July 1994, PP. 10-15.

41. Badra, H. and Mayer, W. " A Neural Network Architecture For Short-term Load Forecasting" *IEEE- International Conference on Neural Network Vol. 7, 1994, PP. 4724-4729.*

42. A. O. Ekwu and B. J. Cory. " Transmission System Expansion Planning by Interactive Method" *IEEE Transaction on Power Apparatus and System. Vol. PAS-103, No.7, July 1984. PP-1583-1591.*

43. SCECO-East and Central Reference Scenario at 2000 and 2010 at Peak Load Developed at June 1995. File # LF6-3-LDVG & LF6-5-LDVG

APPENDICES

1. DC- Load Flow Program for 6-bus system

```
N=6;    % insert the number of buses
M=(N-1);
clear Y H J Jinv x k b tn i j  P dP Bb TR sumr DK inde
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DOING THE ADMITTANCE MATRAIX
% PART # 1
Y= [0 (1+ROW(1))*2.5 0 (1+ROW(2))*1.67 (1+ROW(3))*5.0 0
0 0 (1+ROW(4))*5.0 (1+ROW(5))*2.5 0 (1+ROW(6))*3.33
0 0 0 0 5*(1+ROW(7)) 0
0 0 0 0 0 3.3*(0+ROW(8))
0 0 0 0 0 1.64*(0+ROW(9))
0 0 0 0 0 0 ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% UPDATING THE IMPEDANCE
RR=[0 (1+ROW(1))/.1 0 (1+ROW(2))/.15 (1+ROW(3))/.05 0
0 0 (1+ROW(4))/.05 (1+ROW(5))/.1 0 (1+ROW(6))/.075
0 0 0 0 (1+ROW(7))/.05 0
0 0 0 0 0 (0+ROW(8))/.075
0 0 0 0 0 (0+ROW(9))/.1524
0 0 0 0 0 0 ];
 for vbn=1:N;
  for vbn1=vbn:N;
   if RR(vbn,vbn1)==0;
    R(vbn,vbn1)=0;
    else;
    R(vbn,vbn1)=1/RR(vbn,vbn1);
    end;
 end;
 end;

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % PART# 1.2 COMPLELET THE ADMITTANCE MATRIX
 for i=1:N
   for j=1:N
    Y(j,i)=Y(i,j);
    end
  end
 for i=1:N
   for j=1:N
    if (i==j)
      xy=0;
       else
        Y(i,i)=Y(i,i)+Y(i,j);
        end
     end
  end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STEP #2 TAKING THE VALUES OF SHUNT ADMITTANCE
% OF PI TRANSMISSION LINE
Bb(N,N)=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STEP#3 CALCULATING SELF-ADMITTANCE
for k=1:N;
for l=1:N;
Y(k,k)=Y(k,k)+Bb(k,l);
end
end
for k=1:N;
Y(k,k)=-Y(k,k);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STEP# 3.1 INSERTING THE MAXIMUM CAPACITY POWER VALUES FOR EACH BRANCH
MAX=(1/100)*[0 (1+ROW(1))*100 0 (1+ROW(2))*80 (1+ROW(3))*100 0
0 0 (1+ROW(4))*100 (1+ROW(5))*100 0 (1+ROW(6))*100
0 0 0 0 100*(1+ROW(7)) 0
0 0 0 0 0 100*(0+ROW(8))
0 0 0 0 0 78*(0+ROW(9))
0 0 0 0 0 0 ];
PRICE=[40 60 20 20 40 30 20 30 61];
%%%%%%%%%%
% PART# 3.1 FINDING THE MAGINTUDE OF YBUS
for k=1:N;
        for l=1:N;
        YM (k,l)=abs(Y(k,l));
        end;
        end
YA=angle (Y); % anle of Y bus
%%%%%%%%%%%%%%%%%%%%%%%%
%PART # 3.2 INSERTING THE SPECIFIED VALUES OF REAL POWER GIVEN AT EACH BUS
Ps=1*[-2.4 .85 -1.6 -2.4 5.45];
% insert the given values angle
D=[0.000 .0000 .000 0.000 0.0 0.00];
LOAD=[.3 2.4 1.6 2.4];
P(5)=0;
dP(5)=0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MAIN PROGRAM ITRATION
for q=1:1
%%%%%%%%%%%%%%%%
% PART # 4 CALCULATING  Pcal. AT EACH BUS.

for k=2:N
        for t=1:N
        P(k-1) = P(k-1) +Y(k,t)*(D(k)-D(t));
     end
   dP(k-1)=Ps(k-1)-P(k-1);
        end
```

```
%%%%%%%%%%%%%%
% PART #5 JACOBIAN MATIX
J(M,M)=0;
for k=2:N
        for b=2:N;
        if k==b ;
        x=YM(k,k);
        for tn=1:N;
        J(k-1,b-1)=J(k-1,b-1)+YM(k,tn);
        end
        J(k-1,b-1)=(J(k-1,b-1)-x);
        else
        J(k-1,b-1)= -YM(k,b);
        end
                end
                        end
clear x
%%%%%%%%%%%%%%%%%
% PART# 4.1 FINDING THE INVERSE OF THE JACOBIANE
Jinv=inv(J);
dD=Jinv*dP';
H=D;
%%%%%%%%%%%%%%%%%
% PART # 4.2 UPDATING THE ANGLE VALUES
for i=1:N
    if i==1
        D(i)=0.0;
     else
     D(i)=dD(i-1)+H(i);
    end
  end
clear i


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 5 CHACKING THE STOPPING CRITERIA
g=0;
for k=1:M
        if abs (dD(k)) <=.1
        g=g+1;
        else
        a=0;
        end
        end
clear k
if g==M
Break
else
qa=0;
end
qg=0;
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 6  CHECKING THE CONSTRIANS OF THE CANDIDATE SOLUTION
% a. CHECKING YB-G-D+R=0
inde=0;
DK=Y*D'+[.1 Ps]';
for i=1:N
if (abs(DK(i))>=0 & abs(DK(i))<1)
inde=inde+1;
else;
end ;
end;
if ( inde==N & 0 <= abs(sum(DK)) & sum(LOAD)>=abs(sum(DK)) )
index=0;
else
index=1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% b. CHECKING THE MAXIMUM POWER AND MAXIMUM OF PHASE DIFFERENT
rgh=0;
for isl=1:N
   for jl=isl+1:N
    dphase(isl,jl)=D(isl)-D(jl);
    f(isl,jl) =abs(dphase(isl,jl))*Y(isl,jl);
    rgh=rgh+(f(isl,jl)^2)*R(isl,jl);
    if( dphase(isl,jl)<2*pi/3.14 & f(isl,jl)<=MAX(isl,jl))
      index=index;
      else
      index=index+1;
      end
   end
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 7 CALCULATING THE OBJECTIVE FUNCION
TK=0;
for i=1:cl
TK=TK+PRICE(i)*ROW(i);
end

if index==0
OBJECT(p)=TK+sum(DK)+1000*rgh;
else
OBJECT(p)=TK+sum(DK)+1000+1000*rgh;
end
clear ROW
```

2. DC- Load Flow Program for 25-bus system

```
N=25;   % insert the number of buses
M=(N-1);
clear Y H J Jinv x k b tn i j  P dP Bb TR sumr DK inde
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DOING THE ADMITTANCE MATRAIX
% PART # 1
Y= [0 (1+ROW(1))*92.6 0 0 0 0 (1+ROW(2))*11.5 0 0 0 0 0 (1+ROW(3))*10.6 0 0 0 0 0 0 0 0 0 0 0 0
0 0 50.5*(1+ROW(4)) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(5))*43.3 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(6))*9.6 (1+ROW(7))*7.9 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(8))*11.8 0 0 (1+ROW(9))*11.3 0 0 0 0 (1+ROW(36))*11.08
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(10))*6 0 (1+ROW(11))*6 0 0 0 (1+ROW(12))*16.3 0
0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(13))*21 0 0 (1+ROW(14))*21 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(15))*24 0 0 0 0 0 (1+ROW(16))*25.8 0 0 0
0 0 0 0 0 0 0 0 0 0 (1+ROW(17))*77.5 0 0 0 (1+ROW(18))*69.5 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 (1+ROW(19))*14.8 0 0 0 (1+ROW(20))*9.66 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(21))*40.8 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(22))*19.3 0 0 0 0 0 0 0 0 (1+ROW(23))*11.9 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(24))*11.9 0 (1+ROW(25))*11.9 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(26))*57.8 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(27))*38.6 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(28))*11.9 0  (1+ROW(29))*11.9 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(30))*72 0 0 0 (1+ROW(31))*4.7 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(32))*8.4 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(33))*5.2 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(34))*16.5 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(35))*5.54
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%PART# 000, UPDATING THE IMPEDACE
% PART # 1
RR= [0 (1+ROW(1))/.0027 0 0 0 0 (1+ROW(2))/.0217 0 0 0 0 0 (1+ROW(3))/.0236 0 0 0 0 0 0 0 0 0 0 0 0
0 0 (1+ROW(4))/.005 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(5))/.0058 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(6))/.026 (1+ROW(7))/.0316 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(8))/.0212 0 0 (1+ROW(9))/.0221 0 0 0 0 (1+ROW(36))/.0226
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(10))/.0417 0 (1+ROW(11))/.0417 0 0 0 (1+ROW(12))/.0153 0
0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(13))/.0119 0 0 (1+ROW(14))/.0119 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(15))/.0104 0 0 0 0 0 (1+ROW(16))/.0097 0 0 0
0 0 0 0 0 0 0 0 0 0 (1+ROW(17))/.0032 0 0 0 (1+ROW(18))/.0036 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 (1+ROW(19))/.0169 0 0 0 (1+ROW(20))/.0259 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(21))/.0061 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(22))/.013 0 0 0 0 0 0 0 0 (1+ROW(23))/.0210 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(24))/.021 0 (1+ROW(25))/.021 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(26))/.0043 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(27))/.0065 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(28))/.021 0  (1+ROW(29))/.021 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(30))/.0035 0 0 0 (1+ROW(31))/.0532 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(32))/.0298 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(33))/.0481 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(34))/.0152 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(35))/.0451
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
for vbn=1:N;
 for vbn1=vbn:N;
  if RR(vbn,vbn1)==0;
   R(vbn,vbn1)=0;
   else;
   R(vbn,vbn1)=1/RR(vbn,vbn1);
   end;
end;
end;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART# 1.2 COMPLELET THE ADMITTANCE MATRIX
for i=1:N
  for j=1:N
   Y(j,i)=Y(i,j);
   end
 end
for i=1:N
  for j=1:N
    if (i==j)
      xy=0;
       else
        Y(i,i)=Y(i,i)+Y(i,j);
        end
     end
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% STEP #2  TAKING THE VALUES OF SHUNT ADMITTANCE
% OF PI TRANSMISSION LINE
Bb(N,N)=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% STEP#3 CALCULATING SELF-ADMITTANCE
for k=1:N;
for l=1:N;
```

```
Y(k,k)=Y(k,k)+Bb(k,l);
end
end
for k=1:N;
Y(k,k)=-Y(k,k);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STEP# 3.1 INSERTING THE MAXIMUM CAPACITY POWER VALUES FOR EACH BRANCH
MAX=[0 (1+ROW(1))*8 0 0 0 0 (1+ROW(2))*.65 0 0 0 0 0 (1+ROW(3))*1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 5*(1+ROW(4)) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(5))*2 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(6))*10 (1+ROW(7))*2.5 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(8))*8 0 0 (1+ROW(9))*9.4 0 0 0 0 (1+ROW(36))*2.2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(10))*4.4 0 (1+ROW(11))*2.8 0 0 0 (1+ROW(12))*10.8 0
0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(13))*2.5 0 0 (1+ROW(14))*.9 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(15))*4.9 0 0 0 0 0 (1+ROW(16))*.65 0 0 0
0 0 0 0 0 0 0 0 0 0 (1+ROW(17))*2.6 0 0 0 (1+ROW(18))*2.5 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 (1+ROW(19))*8 0 0 0 (1+ROW(20))*2.5 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(21))*7 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(22))*1 0 0 0 0 0 0 0 0 (1+ROW(23))*.7 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(24))*1 0 (1+ROW(25))*2.5 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(26))*2 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(27))*3.6 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(28))*2.5 0  (1+ROW(29))*5.64 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(30))*4 0 0 0 (1+ROW(31))*3.5 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(32))*1.5 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(33))*1.1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(34))*1.8 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(35))*2.2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
PRICE=[1410 10428 11613 2666 3070 1840 2240 1495 1564 2916 2911 1104 6235 6303 5547 5160 1720
1917 8216 12720 3190 6343 253 253 253 2236 3466 253 253 186 3330 2102 3397 842 1150 644 ];
%%%%%%%%%%
% PART# 3.1 FINDING THE MAGINTUDE OF YBUS
for k=1:N;
        for l=1:N;
        YM (k,l)=abs(Y(k,l));
        end;
        end
YA=angle (Y); % anle of Y bus
%%%%%%%%%%%%%%%%%%%%%%%
%PART # 3.2 INSERTING THE SPECIFIED VALUES OF REAL POWER GIVEN AT EACH BUS
Ps=1*[ -1.34 -1.9 -.78 -.75 -.75 3.16 -2.04 .48 3 4 0 0 -2.9 0 0 -.73 -1.84 -.62 -2.05 -1.43 .5 -1.9 -.71 3.3];
% insert the given values angle
D(N)=0;
LOAD=-1*[-1.34 -1.9 -.78 -.75 -.75 -2.04 -2.9 0 0 -.73 -1.84 -.62 -2.05 -1.43 -1.9 -.71];
P(M)=0;
dP(M)=0;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MAIN PROGRAM ITRATION
for q=1:1
%%%%%%%%%%%%%%%%%%%%%%
% PART # 4 CALCULATING  Pcal. AT EACH BUS.

for k=2:N
        for t=1:N
        P(k-1) = P(k-1) +Y(k,t)*(D(k)-D(t));
    end
   dP(k-1)=Ps(k-1)-P(k-1);
        end

%%%%%%%%%%%%%%%%%%%%%%
% PART #5 JACOBIAN MATIX
J(M,M)=0;
for k=2:N
        for b=2:N;
        if k==b ;
        x=YM(k,k);
        for tn=1:N;
        J(k-1,b-1)=J(k-1,b-1)+YM(k,tn);
        end
        J(k-1,b-1)=(J(k-1,b-1)-x);
        else
        J(k-1,b-1)= -YM(k,b);
        end
                end
                        end
clear x
%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART# 4.1 FINDING THE INVERSE OF THE JACOBIANE
Jinv=inv(J);
dD=Jinv*dP';
H=D;
%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 4.2 UPDATING THE ANGLE VALUES
for i=1:N
    if i==1
        D(i)=0.0;
     else
     D(i)=dD(i-1)+H(i);
    end
   end
clear i
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 5 CHACKING THE STOPPING CRITERIA
g=0;
for k=1:M
        if abs (dD(k)) <=.1
        g=g+1;
        else
```

```
        a=0;
        end
        end
clear k
if g==M
Break
else
qa=0;
end
qg=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 6  CHECKING THE CONSTRIANS OF THE CANDIDATE SOLUTION
% a. CHECKING YB-G-D+R=0
inde=0;
DK=Y*D'+[5.3 Ps]';
for i=1:N
if (abs(DK(i))>=0 & abs(DK(i))<10)
inde=inde+1;
else;
end ;
end;
if ( inde==N & 0 < abs(sum(DK)) & sum(LOAD)>abs(sum(DK)) )
index=0;
else
index=1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% b. CHECKING THE MAXIMUM POWER AND MAXIMUM OF PHASE DIFFERENT
rhg=0;
for il=1:N
  for jl=il+1:N
   dphase(il,jl)=D(il)-D(jl);
   f(il,jl) =abs(dphase(il,jl))*Y(il,jl);
    rhg=rhg+(f(il,jl)^2)*R(il,jl);
    if( dphase(il,jl)<2*pi/3.14 & f(il,jl)<=MAX(il,jl))
    index=index;
    else
    index=index+1;
    end
  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 7 CALCULATING THE OBJECTIVE FUNCION
TK=0;
for i=1:cl
TK=TK+PRICE(i)*ROW(i);
end

if index==0
OBJECT(p)=TK+sum(DK)+K*rhg;
```

```
else
OBJECT(p)=TK+sum(DK)+K*rhg+999999;
end
clear ROW
```

3. DC- Laod Flow for the SCECO-EAST System

```
K=10000;
N=14;   % insert the number of buses
M=(N-1);
clear Y H J Jinv x k b tn i j  P dP Bb TR sumr DK inde
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DOING THE ADMITTANCE MATRAIX
% PART # 1
Y= [0 (1+ROW(1))*34.2 0 0 0 0 0 0 0 0 0 0 0 0
0 0 55.45*(2+ROW(2)) 0 108.2*(1+ROW(3)) 0 0 0 0 0 0 0 0 0
0 0 0 (2+ROW(4))*19.4 (1+ROW(5))*62.86 0 (2+ROW(6))*25.84 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 (2+ROW(7))*116.3 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 (3+ROW(8))*93.02 (2+ROW(9))*93.02 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 (2+ROW(10))*51.70 0 0 0 0
0 0 0 0 0 0 0 0 0 0 (2+ROW(11))*93.02 (1+ROW(12))*73.74 (1+ROW(13))*13.20 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(14))*55.31
0 0 0 0 0 0 0 0 0 0 0 0 0 0];
%%%%%%%%%%%%%%%%%%%%%%%%%%
RR= [0 (1+ROW(1))/.0072 0 0 0 0 0 0 0 0 0 0 0 0
0 0 (2+ROW(2))/.0045 0 (1+ROW(3))/.0023 0 0 0 0 0 0 0 0 0
0 0 0 (2+ROW(4))/.0129 (1+ROW(5))/.004 0 (2+ROW(6))/.0097 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 (2+ROW(7))/.0022 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 (3+ROW(8))/.0027 (2+ROW(9))*.0027 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 (2+ROW(10))/.0048 0 0 0 0
0 0 0 0 0 0 0 0 0 0 (2+ROW(11))/.0027 (1+ROW(12))/.0034 (1+ROW(13))/.0189 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(14))/.0045
0 0 0 0 0 0 0 0 0 0 0 0 0 0];

 for vbn=1:N;
  for vbn1=vbn:N;
   if RR(vbn,vbn1)==0;
    R(vbn,vbn1)=0;
    else;
    R(vbn,vbn1)=1/RR(vbn,vbn1);
    end;
 end;
 end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART# 1.2 COMPLELET THE ADMITTANCE MATRIX
```

```
for i=1:N
  for j=1:N
   Y(j,i)=Y(i,j);
   end
 end
for i=1:N
  for j=1:N
    if (i==j)
      xy=0;
       else
        Y(i,i)=Y(i,i)+Y(i,j);
        end
     end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% STEP #2 TAKING THE VALUES OF SHUNT ADMITTANCE
% OF PI TRANSMISSION LINE
Bb(N,N)=0;
%Bb(17,19)=4.33;Bb(19,17)=4.33;Bb(17,23)=35;Bb(23,17)=35;Bb(5,17)=87;Bb(17,5)=35;Bb(4,19)=58;B
b(19,4)=58;
%Bb(21,19)=38.5;Bb(19,21)=38.5;Bb(23,18)=62;Bb(18,23)=62;Bb(4,18)=71;Bb(18,4)=71;Bb(5,20)=83.5;
Bb(20,5)=83.5;
%Bb(20,21)=.814;Bb(21,20)=.814;Bb(14,12)=18.3;Bb(12,14)=18.3;Bb(24,6)=120.4;Bb(6,24)=120.3;Bb(6,
18)=45;Bb(18,6)=45;
%Bb(6,20)=45;Bb(20,6)=45;Bb(16,7)=22.3;Bb(7,16)=22.3;Bb(8,16)=11.4;Bb(16,8)=11.4;Bb(7,13)=22.3;
Bb(13,7)=22.3;
%Bb(1,13)=10;Bb(13,1)=10;Bb(1,7)=11;Bb(7,1)=11;Bb(8,22)=24.5;Bb(22,8)=24.5;Bb(14,22)=55;Bb(22,1
4)=55;Bb(11,14)=10;Bb(14,11)=10;
%Bb(15,22)=37;Bb(22,15)=37;Bb(3,22)=41;Bb(22,3)=41;Bb(9,15)=65.8;Bb(15,9)=65.8;Bb(15,10)=9;Bb(
10,15)=9;Bb(9,11)=18.3;Bb(11,9)=18.3;
%Bb(1,2)=22;Bb(2,1)=22;Bb(10,11)=14;Bb(11,10)=14;Bb(2,3)=12;Bb(3,2)=12;Bb(1,25)=166.7;Bb(25,1)=
166.7;Bb(5,25)=333.3;Bb(25,5)=333.3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STEP#3 CALCULATING SELF-ADMITTANCE
for  k=1:N;
for l=1:N;
Y(k,k)=Y(k,k)+Bb(k,l);
end
end
for k=1:N;
Y(k,k)=-Y(k,k);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STEP# 3.1 INSERTING THE MAXIMUM CAPACITY POWER VALUES FOR EACH BRANCH
MAX=[0 (1+ROW(1))*11.5 0 0 0 0 0 0 0 0 0 0 0 0
0 0 11.5*(2+ROW(2)) 0 11.5*(1+ROW(3)) 0 0 0 0 0 0 0 0 0
0 0 0 (2+ROW(4))*11.5 (1+ROW(5))*11.5 0 (2+ROW(6))*11.5 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 (2+ROW(7))*11.5 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 (3+ROW(8))*11.5 (2+ROW(9))*11.5 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 (2+ROW(10))*11.5 0 0 0 0
0 0 0 0 0 0 0 0 0 0 (2+ROW(11))*11.5 (1+ROW(12))*11.5 (1+ROW(13))*11.5 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 (1+ROW(14))*11.5
0 0 0 0 0 0 0 0 0 0 0 0 0 0];
PRICE=[29349 18127 9279 51792 15969 38844 8632 10790 10790 19422 10790 12948 72293 17264 ];
%%%%%%%%
% PART# 3.1 FINDING THE MAGINTUDE OF YBUS
for k=1:N;
        for l=1:N;
        YM (k,l)=abs(Y(k,l));
        end;
        end
YA=angle (Y); % anle of Y bus
%%%%%%%%%%%%%%%%%%%%%%%
%PART # 3.2 INSERTING THE SPECIFIED VALUES OF REAL POWER GIVEN AT EACH BUS
%Ps=1*[25.5 -10.2 -8.7 -15.0 18.8 0 25.5 -22.0 0 -8.0 19.0 -3 -4 ];
Ps=1*[22.8 -9.68 -8.20 -13.33 16.82 0 22.8 -19.60 0 -7.64 17.10 -2.30 -2.41];
% insert the given values angle
D(N)=0;
LOAD=-1*[-17.9  -10.2 -8.7 -15.0 -22.0  -8.0  -3 -4 ];
P(M)=0;
dP(M)=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MAIN PROGRAM ITRATION
for q=1:1
%%%%%%%%%%%%%%%%%%
% PART # 4 CALCULATING  Pcal. AT EACH BUS.

for k=2:N
         for t=1:N
         P(k-1) = P(k-1) +Y(k,t)*(D(k)-D(t));
     end
   dP(k-1)=Ps(k-1)-P(k-1);
         end

%%%%%%%%%%%%%%%%%%
% PART #5 JACOBIAN MATIX
J(M,M)=0;
for k=2:N
        for b=2:N;
        if k==b ;
        x=YM(k,k);
        for tn=1:N;
        J(k-1,b-1)=J(k-1,b-1)+YM(k,tn);
        end
        J(k-1,b-1)=(J(k-1,b-1)-x);
        else
```

```
        J(k-1,b-1)= -YM(k,b);
        end
              end
                    end
clear x
%%%%%%%%%%%%%%%%%%%
% PART# 4.1 FINDING THE INVERSE OF THE JACOBIANE
Jinv=inv(J);
dD=Jinv*dP';
H=D;
%%%%%%%%%%%%%%%%%%%%
% PART # 4.2 UPDATING THE ANGLE VALUES
for i=1:N
   if i==1
      D(i)=0.0;
    else
    D(i)=dD(i-1)+H(i);
   end
  end
clear i
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 5 CHACKING THE STOPPING CRITERIA
g=0;
for k=1:M
      if abs (dD(k)) <=.1
      g=g+1;
      else
      a=0;
      end
      end
clear k
if g==M
Break
else
qa=0;
end
qg=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 6  CHECKING THE CONSTRIANS OF THE CANDIDATE SOLUTION
% a. CHECKING YB-G-D+R=0
inde=0;
DK=Y*D'+[-16.36 Ps]';
for i=1:N
if (abs(DK(i))>=0 & abs(DK(i))<10)
inde=inde+1;
else;
end ;
end;
if ( inde==N & 0 < abs(sum(DK)) & sum(LOAD)>abs(sum(DK)) )
index=0;
else
```

```
index=1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%
% b. CHECKING THE MAXIMUM POWER AND MAXIMUM OF PHASE DIFFERENT
rgh=0;
for il=1:N
  for jl=il+1:N
   dphase(il,jl)=D(il)-D(jl);
   f(il,jl) =abs(dphase(il,jl))*Y(il,jl);
   rgh=rgh+(f(il,jl)^2)*R(il,jl);
     if( dphase(il,jl)<2*pi/3.14 & f(il,jl)<=MAX(il,jl))
     index=index;
     else
     index=index+1;
     end
  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART # 7 CALCULATING THE OBJECTIVE FUNCION
TK=0;
for i=1:cl
TK=TK+PRICE(i)*ROW(i);
end

if index==0
OBJECT(p)=TK+sum(DK)+K*rgh;
else
OBJECT(p)=TK+sum(DK)+1000000+K*rgh;
end
clear ROW
```

4. Genetic Algorithm Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This program applies a Genetic Algorithm to six bus system
% Parameters of the Program are as follows:
% 1. Population Size (ps) and it is set to be 15
% 2. Number of right-of-way (cl) and it is set to be 9
% 3. Generation number (GN) and it is set to be 1200
% 4. Cross-Over rate (pcross)and it is set to be 0.82
% 5. Mutation rate (pmute) and it is set to be 0.071
% 6. Degits Number of The Binary Number (BN) and
%    it is set to be 2
% 7. The Starting Global Minimum of The Objective
%    Function (min) it is set to 999999
%
% These parameters are set for six-bus system, If the
% users want to change the system, they should change the
% load-flow program lfax in this program to ld2x (for
% IEEE-25 system) or lfsx for SCECO-East system. Also the
% parameters specifed above should be changed to fit each system as
% shown in the thesis.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Initial The Parameter of The Program
%Population Size
ps=15;
% number of right-of-way
cl=9;
% Generation number
GN=1200;
% Cross-Over rate
pcross=.82 ;
% Mutation rate
pmute=.071;
% Degits Number of The Binary Number
BN=2;
% The Global Minimum of The Objective Function
min=999999;
%Initial population
u=4*rand(ps,cl);
for i=1:ps
   for j=1:cl
      for ii=0:10;
      if u(i,j)>ii&u(i,j)<ii+1;
      u(i,j)=ii;
      break
      else
      end
      end
```

```
  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION
 for jmain=1:GN;
    for p=1:ps;
      for ia=1:cl;
      ROW(ia)=u(p,ia);
      end;
    lfax
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
 for k=1:ps-1;
  for L=k+1,ps;
    if (OBJECT(k)>OBJECT(L));
    TEMP = OBJECT(k);
    OBJECT(k) = OBJECT(L);
    OBJECT(L) = TEMP;
    for jm =1:cl;
    TEM = u(k, jm);
    u(k,jm) = u(L,jm);
    u(L,jm) = TEM;
    end;
    else;
    end;
  end;
 end;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%
 % SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
 top(jmain)=OBJECT(1);
 for ie=1:cl;
 BROW(jmain,ie)=u(1,ie);
 end
 BROW(jmain,cl+1)=OBJECT(1);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%
 % NEW GENERATION PRODUCTION BY FOLLOWING
 % 1.CROSSOVER SELECTION USING ROLETTE WHEEL SELECTION
  for j=1:ps;
   fit(j)=1/(1+OBJECT(j));
  end ;
 sumfit=sum(fit);
 %%%%%%%%%%%%%%%%%%%%%
 for Ge=1:2:(ps-2)   % main loop of crossover implementation
  for m=1:2;
  partsum=0;
  Ran=rand*sumfit;
   for j=1:ps;
```

```
  partsum= partsum+fit(j);
  if (partsum >= Ran| j==ps);
  s(m)=j;
  break
  else
  end
  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% CROSSOVER IMPLEMENTATION

   if(pcross>rand)
   as(1)=cl* rand;
   as(2)=cl* rand;
          for ss=1:2;
             for ii=0:100;
        if as(ss)>ii & as(ss)<ii+1;
        as(ss)=ii+1;
        break
        else
        end
       end
     end
   else
   as(1)=1;
   as(2)=cl;
   end
 sort(as);

  for ic =1:cl;
   Parent1 (ic) = u(s(1),ic);
   Parent2 (ic) = u(s(2),ic);
  end

  for jc = 1:as(1)
 child1(jc)=Parent1(jc);
 child2(jc)=Parent2(jc);
  end

  for jd = as(1)+1: as(2)
   child1(jd) =Parent2(jd);
   child2(jd) =Parent1(jd);
  end

  for  jc = as(2)+1:cl;
 child1(jc) = Parent1(jc);
 child2(jc) = Parent2(jc);
  end

  for ig=1:cl
 W(Ge,ig)=child1(ig);
```

```
W(Ge+1,ig)=child2(ig);
 end

clear child1 child2 Parent2 Parent1

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% MUTATION IMPLEMENTATION

for it=1:ps-2;
  if(pmute>rand);
  Is(1)= cl*rand;
  Is(2)= BN*rand;
    for ss= 1:2;
     for ii=0:100;
          if Is(ss)>ii & Is(ss)<ii+1;
          Is(ss)=ii+1;
          break;
          else;
          end;
       end;
     end;

 IM=W(it,Is(1));
    for     jr=BN:-1:1;
      xx=IM/2;
     for i=0:100;
      if ((xx>i & xx<i+1) | xx == i);
      IM=i;
      R=xx-IM;
      break;
      else;
      end;
     end;
    if R==0;
    X(jr)=0;
    else;
    X(jr)=1;
    end;
   end;
  clear xx


    if (X(Is(2))==1)
    X(Is(2))=0;
    else
    X(Is(2))=1;
    end
    accum = 0;
    po2 = 1;
```

```
for jw =BN:-1:1;
 if X(jw)==1;
 accum = accum +po2;
 Po2=Po2*2;
 else;
 Po2 = Po2*2;
 end;
 end ;
 W(it,Is(1))=accum;
 else
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
% NEW GENERATION ARRANGMENT
 for iq=3:ps;
  for iu=1:cl;
  u(iq,iu)=W(iq-2,iu);
  end
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
% STOPPING CRITERIA CHECK

  E(jmain)=min-top(jmain);
   if top(jmain)<min;
   min=top(jmain);
for tre=1:cl;
OPTIMUM(tre) =BROW(jmain,tre);
end
   else
   end

 end   %MAIN LOOP OF THE PROGRAM

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%
```

5. Tabu Search Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program applies a Genetic Algorithm to six bus system
% Parameters of the Program are as follows:
% 1. Movement number (mn) and it is set to be 4
% 2. Number of right-of-way (cl) which is set to be 9 to solve
%    the six-bus system
% 3. Total itration number (GN) and it is set to be 1000
% 4. Size OF tabu list (tl) and it is set to be 9;
% 5. The Starting Global Minimum of The Objective
%    Function (min) it is set to 999999
%
% These parameters are set for six-bus system. If the
% users want to change the system, they should change the
% load-flow program lfax in this program to ld2x (for
% IEEE-25 system) or lfsx for SCECO-East system. Also the
% parameters specifed above should be changed to fit each system as
% shown in the thesis.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Initial The Parameter of The Program
% MOVEMENT NUMBER
mn=4;
% number of right-of-way
cl=9;
% Generation number
GN=1000;
% The Global Minimum of The Objective Function
min=999999;
% SIZE OF TABU LIST
tl=9;
% INTIAL VALUES OF TABU LIST
xtl=rand(tl,cl+1);
%Initial population
u=4*rand(mn,cl);
for i=1:mn
   for j=1:cl
      for ii=0:10;
      if u(i,j)>ii&u(i,j)<ii+1;
      u(i,j)=ii;
      break
      else
      end
      end
   end
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION
 for jmain=1:GN;
    for p=1:mn;
      for ia=1:cl;
      ROW(ia)=u(p,ia);
      end;
   Ifax
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
 for k=1:mn-1;
  for L=k+1,mn;
    if (OBJECT(k)>OBJECT(L));
    TEMP = OBJECT(k);
    OBJECT(k) = OBJECT(L);
    OBJECT(L) = TEMP;
    for jm =1:cl;
    TEM = u(k, jm);
    u(k,jm) = u(L,jm);
    u(L,jm) = TEM;
    end;
    else;
    end;
  end;
 end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
top(jmain)=OBJECT(1);
for ie=1:cl;
BROW(jmain,ie)=u(1,ie);
end
BROW(jmain,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CHECKING THE ACCEPTENCY TO THE TABU LIST
GR=0;S=0;
 for i=1:tl
  index=0;
    for j=1:cl
      if xtl(i,j)==BROW(jmain,j);
      index=index+1;
      else
      end
     end
      if index==(cl)
      S=S+1;
      else
```

```
  GR=GR+1;
  end
end

if GR==tl
   for i=tl:-1:2
      for j=1:cl+1
      xtl(i,j)=xtl(i-1,j);
      end
    end
   for j=1:cl+1
   xtl(1,j)=BROW(jmain,j);
   end
  else
  end
clear GR S
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% GENERATE NEW RANDOM MOVEMENT
%%%%%%%%%%%%%%%%%%%%%
% PART(2):GENERATE INTEGER RANDOM VECTOR
urc=4*rand(mn,1);
for i=1:mn
 for ii=0:100;
 if urc(i,1)>ii & urc(i,1)<ii+1
 urc(i,1)=ii;
 break
 else
 end
 end
end
%%%%%%%%%%%%%%%%%%%%%
% PART(3): INSER THE RANDOM VECTOR TO THE MATRIX THAT GENERATED FROM
PART(1)
for xt=1:mn
  xcl=cl*rand;
  for ii=0:100;
  if xcl>ii & xcl<ii+1
  xcl=ii+1;
  break
  else
  end
  end
 u(xt,xcl)=urc(xt,1);
 end
 %%%%%%%%%%%%%%%%%%%%%%
 % PART(4):CHANGE SOME STATE BY RANDOM CHOICE
 % A.SUBSTRCTION
 if rand<.91
 for imain=1:3;
 jj=cl*rand;
  for ii=0:100;
```

```
  if jj>ii & jj<ii+1
  jj=ii+1;
  break
  else
  end
  end
TQ=mn*rand;
 for ii=0:100;
  if TQ>ii & TQ<ii+1
  TQ=ii+1;
  break
  else
  end
  end

aw=u(TQ,jj);
if aw==0;
u(TQ,jj)=aw;
else
u(TQ,jj)=aw-1;
end
 end
else
end
%%%%%%%%%%%%%%%%%%%%%%
%B.ADDITION
if rand<.055
for imain=1:1;
jj=cl*rand;
  for ii=0:100;
   if jj>ii & jj<ii+1
   jj=ii+1;
   break
   else
   end
   end
  TQ=mn*rand;
   for ii=0:100;
    if TQ>ii & TQ<ii+1
    TQ=ii+1;
    break
    else
    end
    end

  aw=u(TQ,jj);
  if aw==3;
  u(TQ,jj)=aw;
  else
  u(TQ,jj)=aw+1;
  end
   end
```

```
else
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% STOPPING CRITEREIA CHECK
  E(jmain)=min-top(jmain);
   if top(jmain)<min;
   min=top(jmain);;
   for sss=1:cl
   OPTIMUM(sss)=BROW(jmain,sss);
   end
   else
   end

 end   %MAIN LOOP OF THE PROGRAM

 clear OBJECT
 clear OBJECT
 p=1;cl=9;
 ROW=OPTIMUM;
 lfa
 min=OBJECT-K*rgh+30;
 OPTIMUM(6)=OPTIMUM(6)+1;
```

6. Artificail Neural Network Program

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
%This program applies a Neural Network to six bus system
% Parameters of the Program are as follows:
% 1. State Space Size, training set, (ps) and it is set to be 50
% 2. Right-of-way (cl) and it is set to be 9
% 3. Total Iteration number (GN1)and it is set to be 1000
% 4. number of the generation in each iteration (pl1) and it is set to be 10
% 5. Input neuron is one, hidden nuerons are three and output neurons are
%   same as the size of (cl) which in this case is 9
% 6. The Starting Global Minimum of The Objective
%    Function (min) it is set to 999999
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
% Samlpe of the ANN implementation in the program
%
%**** loop for getting the new soluion states
%**** for rp=1:pl1
%**** [w1,b1,w2,b2]=initff(st,4,'logsig',T,'purelin');
%**** [w1,b1,w2,b2,epoch,tr]=trainbpx(w1,b1,'logsig',w2,b2,'purelin',st,T,tp);
%**** d9=K2*.81*min*rand;
%**** s=(1.8/K1)*simuff(d9,w1,b1,'logsig',w2,b2,'purelin');
%**** end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
% These parameters are set for six-bus system. If the
% users want to change the system, they should change the
% load-flow program lfax in this program to ld2x (for
% IEEE-25 system) or lfsx for SCECO-East system. Also the
% parameters specifed above should be changed to fit each system as
% shown in the thesis.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GLOBAL MINIMUM VALUES
K1=.0251;K2=1;
min=999999;
% STATE SPACE
ps=50;
% RIGHT OF WAY
cl=9;
%ITERATION
GN1=1000;
pl1=10;
pss=50;
%INITIAL SOLUTION STATES
u=4*rand(pss,cl);
for i=1:pss
  for j=1:cl
    for ii=0:10;
```

```
        if u(i,j)>ii&u(i,j)<ii+1;
        u(i,j)=ii;
        break
        else
        end
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION FOR
% THE INTIAL SOLUTION STATES
        for p=1:pss;
        for ia=1:cl;
        ROW(ia)=u(p,ia);
        end;
    lfax
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
for www=1:pss
for k=1:pss-1;
 for L=k+1,pss;
   if (OBJECT(k)>OBJECT(L));
   TEMP = OBJECT(k);
   OBJECT(k) = OBJECT(L);
   OBJECT(L) = TEMP;
   for jm =1:cl;
   TEM = u(k, jm);
   u(k,jm) = u(L,jm);
   u(L,jm) = TEM;
   end;
   else;
   end;
 end;
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
top(1)=OBJECT(1);
for ie=1:cl;
BROW(1,ie)=u(1,ie);
end
BROW(1,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STORING THE BEST SOLUTION
   if top(1)<min;
   min=top(1);
   else
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% APPLYING NEURAL NETWORK APPROACH TO HAVE A NEW SOLUTION STATES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% PART ONE GIVING THE TRAINING STATES
for qqq1=1:ps
 for qqq2=1:cl
T(qqq2,qqq1)=K1*u(qqq1,qqq2);
end
end
for qqq1=1:ps
st(qqq1)=K2*OBJECT(qqq1);
end
clear OBJECT u
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 for jmain=2:GN1
 gt1=[T' st'];

 tp=[100 1000 .6 .03];
 % loop for getting the new soluion states
 for rp=1:pl1
 [w1,b1,w2,b2]=initff(st,4,'logsig',T,'purelin');
 [w1,b1,w2,b2,epoch,tr]=trainbpx(w1,b1,'logsig',w2,b2,'purelin',st,T,tp);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 d9=K2*.81*min*rand;
 s=(1.8/K1)*simuff(d9,w1,b1,'logsig',w2,b2,'purelin');
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 for imain=1:1;
 TQ1=cl*rand;
  for ii=0:100;
   if TQ1>ii & TQ1<ii+1
   TQ1=ii+1;
   break
   else
   end
   end
  end
  sr=abs(s'/1);

    for j=1:cl
        for ii= 0:20;
        if sr(j)>ii&sr(j)<ii+1;
        sr(j)=ii;
        break
        else
        end
        end
      end
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   for j=1:cl
   ns(rp,j)=sr(j);
   end
   end
   %%%%%%%%%%%%%%%%%%%%%%%%
   % STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION FOR
```

```
% THE INTIAL SOLUTION STATES
      for p=1:pl1;
      for ia=1:cl;
      ROW(ia)=ns(p,ia);
      end;
   lfax
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
for www=1:pl1
for k=1:pl1-1;
 for L=k+1,pl1;
   if (OBJECT(k)>OBJECT(L));
   TEMP = OBJECT(k);
   OBJECT(k) = OBJECT(L);
   OBJECT(L) = TEMP;
   for jm =1:cl;
   TEM = ns(k, jm);
   ns(k,jm) = ns(L,jm);
   ns(L,jm) = TEM;
   end;
   else;
   end;
  end;
 end;
 end;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
 top(jmain)=OBJECT(1);
 for ie=1:cl;
 BROW(jmain,ie)=ns(1,ie);
 end
 BROW(jmain,cl+1)=OBJECT(1);
 ns=ns;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % STORING THE BEST SOLUTION
    if top(jmain)<min;
    min=top(jmain);
    for mx=1:cl;
    OPT(mx)=BROW(jmain,mx);
    end
    else
    end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  gt2=[ns OBJECT'];
  gt=[gt1;gt2];
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
  for www=1:(pl1+ps)
  for k=1:(pl1+ps)-1;
   for L=k+1,(pl1+ps);
     if (gt(k,cl+1)>gt(L,cl+1));
```

```
TEMP = gt(k,cl+1);
gt(k,cl+1) = gt(L,cl+1);
gt(L,cl+1) = TEMP;
for jm =1:cl+1;
TEM = gt(k, jm);
gt(k,jm) = gt(L,jm);
gt(L,jm) = TEM;
end;
else;
end;
end;
end;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:ps
for jj=1:cl
u(ii,jj)=gt(ii,jj);
end
end
T=K1*u';
%%%%%%%%%%%%
for ii=1:ps
st(ii)=K2*gt(ii,cl+1);
end
%main loop
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE BEST MOVE FROM THE NEURAL NETWORK
for ww=1:GN1
Bm(ww)=BROW(ww,cl+1);
end
for op=1:GN1
for k=1:GN1-1;
for L=k+1,GN1;
if (Bm(k)>=Bm(L));
TEMP = Bm(k);
Bm(k) = Bm(L);
Bm(L) = TEMP;
for jm =1:cl+1;
TEM = BROW(k, jm);
BROW(k,jm) = BROW(L,jm);
BROW(L,jm) = TEM;
end;
else;
end;
end;
end;
end;
```

7. First hybrid method between Tabu and Genetic

```
% This program applies a the first hybrid model of Tabu Search with Genetic Algorithm to
% six bus system parameters of the program are as follows:
%
% A. Tabu Parameters
% 1. Movement number (mn) and it is set to be 4
% 2. Number of right-of-way (cl) which is set to be 9 to solve
%    the six-bus system
% 3. Total itration number (GN) and it is set to be 1000
% 4. Size OF tabu list (tl) and it is set to be 9;
% 5. The Starting Global Minimum of The Objective
%    Function (min) it is set to 999999
%
% B. Genetic Algorithm
% 1. Population Size (ps) and it is set to be 15
% 2. Number of right-of-way (cl) and it is set to be 9
% 3. Generation number (GN) and it is set to be 1200
% 4. Cross-Over rate (pcross)and it is set to be 0.82
% 5. Mutation rate (pmute) and it is set to be 0.071
% 6. Degits Number of The Binary Number (BN) and
%    it is set to be 2
%
% These parameters are set for six-bus system, If the
% users want to change the system, they should change the
% load-flow program lfax in this program to ld2x (for
% IEEE-25 system) or lfsx for SCECO-East system. Also the
% parameters specifed above should be changed to fit each system as
% shown in the thesis.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Initial The Parameter of The Program
% MOVEMENT NUMBER
mn=4;
% number of right-of-way
cl=9;
% Generation number
GN1=1000;
% The Global Minimum of The Objective Function
min=999999;
% SIZE OF TABU LIST
tl=9;
% INTIAL VALUES OF TABU LIST
xtl=rand(tl,cl+1);
%Initial population
u=4*rand(mn,cl);
for i=1:mn
   for j=1:cl
      for ii=0:10;
```

```
      if u(i,j)>ii&u(i,j)<ii+1;
      u(i,j)=ii;
      break
      else
      end
      end
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION
 for jmain=1:GN1;
     for p=1:mn;
       for ia=1:cl;
       ROW(ia)=u(p,ia);
       end;
    lfax
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS

 for e1=1:mn
 for  k=1:mn-1;
  for  L=k+1,mn;
    if (OBJECT(k)>OBJECT(L));
    TEMP = OBJECT(k);
    OBJECT(k) = OBJECT(L);
    OBJECT(L) = TEMP;
    for jm =1:cl;
    TEM = u(k, jm);
    u(k,jm) = u(L,jm);
    u(L,jm) = TEM;
    end;
    else;
    end;
  end;
 end;
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
top(jmain)=OBJECT(1);
for ie=1:cl;
BROW(jmain,ie)=u(1,ie);
end
BROW(jmain,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CHECKING THE ACCEPTENCY TO THE TABU LIST
GR=0;S=0;
 for i=1:tl
```

```
index=0;
 for j=1:cl
  if xtl(i,j)==BROW(jmain,j);
  index=index+1;
  else
  end
 end
  if index==(cl)
  S=S+1;
  else
  GR=GR+1;
  end
end

if GR==tl
   for i=tl:-1:2
      for j=1:cl+1
      xtl(i,j)=xtl(i-1,j);
      end
    end
   for j=1:cl+1
   xtl(1,j)=BROW(jmain,j);
   end
 else
 end
clear GR S
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% PART(2):GENERATE INTEGER RANDOM VECTOR
urc=4*rand(mn,1);
for i=1:mn
 for ii=0:100;
 if urc(i,1)>ii & urc(i,1)<ii+1
 urc(i,1)=ii;
 break
 else
 end
 end
end
%%%%%%%%%%%%%%%%%%%%%%%%
% PART(3): INSERT THE RANDOM VECTOR TO THE MATRIX THAT GENERATED FROM
PART(1)
for xt=1:mn
  xcl=cl*rand;
  for ii=0:100;
  if xcl>ii & xcl<ii+1
  xcl=ii+1;
  break
  else
  end
  end
 u(xt,xcl)=urc(xt,1);
```

```
end
%%%%%%%%%%%%%%%%%%%%%%%
% PART(4):CHANGE SOME STATE BY RANDOM CHOICE
% A.SUBSTRCTION
if rand<.91
for imain=1:3;
jj=cl*rand;
  for ii=0:100;
   if jj>ii & jj<ii+1
   jj=ii+1;
   break
   else
   end
   end
 TQ=mn*rand;
  for ii=0:100;
   if TQ>ii & TQ<ii+!
   TQ=ii+1;
   break
   else
   end
   end

 aw=u(TQ,jj);
 if aw==0;
 u(TQ,jj)=aw;
 else
 u(TQ,jj)=aw-1;
 end
  end
 else
 end
 %%%%%%%%%%%%%%%%%%%%%%%
 %B.ADDITION
 if rand<.055
 for imain=1:1;
 jj=cl*rand;
   for ii=0:100;
    if jj>ii & jj<ii+1
    jj=ii+1;
    break
    else
    end
    end
  TQ=mn*rand;
   for ii=0:100;
    if TQ>ii & TQ<ii+1
    TQ=ii+1;
    break
    else
    end
    end
```

```
aw=u(TQ,jj);
if aw==3;
u(TQ,jj)=aw;
else
u(TQ,jj)=aw+1;
end
 end
else
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% STOPPING CRITEREIA CHECK
  E(jmain)=min-top(jmain);
   if top(jmain)<min;
   min=top(jmain);;
   for sss=1:cl
   OPTIMUM(sss)=BROW(jmain,sss);
   end
   else
   end

  % if abs(E(jmain))< .01;
  % break
  % else
  % end
 end   %MAIN LOOP OF THE PROGRAM



 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % SORTING THE BEST MOVE FROM THE TABU SEARCH
 for e1=1:GN1
 for  k=1:GN1-1;
  for  L=k+1,GN1;
    if (BROW(k,cl+1)>BROW(L,cl+1));
    for jm =1:cl+1;
    TEMP = BROW(k,jm);
    BROW(k,jm) = BROW(L,jm);
    BROW(L,jm) = TEMP;
    end
    end;
  end;
 end;
 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % GENETIC ALGORITHM
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %Initial The Parameter of The Program
 %Population Size
 ps=15;
 % number of right-of-way
```

```
cl=9;
% Generation number
GN2=1200;
% Cross-Over rate
pcross=.82 ;
% Mutation rate
pmute=.07;
% Degits Number of The Binary Number
BN=2;
%New population for the GA
for ww=1:ps
   for qq=1:cl
     u(ww,qq)=BROW(ww,qq);
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION
 for jmain=1:GN2;
     for p=1:ps;
       for ia=1:cl;
       ROW(ia)=u(p,ia);
       end;
    lfax
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
for e1=1:ps
for  k=1:ps-1;
  for  L=k+1,ps;
     if (OBJECT(k)>OBJECT(L));
     TEMP = OBJECT(k);
     OBJECT(k) = OBJECT(L);
     OBJECT(L) = TEMP;
     for jm =1:cl;
     TEM = u(k, jm);
     u(k,jm) = u(L,jm);
     u(L,jm) = TEM;
     end;
     else;
     end;
  end;
 end;
 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%
 % SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
 top(jmain)=OBJECT(1);
 for ie=1:cl;
 BROWg(jmain,ie)=u(1,ie);
 end
```

```
BROWg(jmain,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% NEW GENERATION PRODUCTION BY FOLLOWING
% 1.CROSSOVER SELECTION USING ROLETTE WHEEL SELECTION
 for j=1:ps;
  fit(j)=1/(1+OBJECT(j));
 end ;
sumfit=sum(fit);
%%%%%%%%%%%%%%%%%%%%
for Ge=1:2:(ps-2)   % main loop of crossover implementation
 for m=1:2;
 partsum=0;
 Ran=rand*sumfit;
   for j=1:ps;
   partsum= partsum+fit(j);
   if (partsum >= Ran| j==ps);
   s(m)=j;
   break
   else
   end
   end
 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%
 % CROSSOVER IMPLEMENTATION

     if(pcross>rand)
     as(1)=cl* rand;
     as(2)=cl* rand;
            for ss=1:2;
              for ii=0:100;
         if as(ss)>ii & as(ss)<ii+1;
         as(ss)=ii+1;
         break
         else
         end
        end
       end
     else
     as(1)=1;
     as(2)=cl;
     end
   sort(as);

    for ic =1:cl;
     Parent1 (ic) = u(s(1),ic);
     Parent2 (ic) = u(s(2),ic);
    end

    for jc = 1:as(1)
   child1(jc)=Parent1(jc);
```

```
child2(jc)=Parent2(jc);
 end

 for jd = as(1): as(2)
  child1(jd) =Parent2(jd);
  child2(jd) =Parent1(jd);
 end

 for  jc = as(2):cl;
child1(jc) = Parent1(jc);
child2(jc) = Parent2(jc);
 end

 for ig=1:cl
 W(Ge,ig)=child1(ig);
 W(Ge+1,ig)=child2(ig);
 end

 clear child1 child2 Parent2 Parent1

 end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% MUTATION IMPLEMENTATION

 for it=1:ps-2;
  if(pmute>rand);
  Is(1)= cl*rand;
  Is(2)= BN*rand;
    for ss= 1:2;
     for ii=0:100;
          if Is(ss)>ii & Is(ss)<ii+1;
          Is(ss)=ii+1;
          break;
          else;
          end;
      end;
    end;

  IM=W(it,Is(1));
    for      jr=BN:-1:1;
      xx=IM/2;
     for i=0:100;
      if ((xx>i & xx<i+1) | xx == i);
      IM=i;
      R=xx-IM;
      break;
      else;
      end;
     end;
    if R==0;
```

```
  X(jr)=0;
  else;
  X(jr)=1;
  end;
 end;
clear xx


 if (X(Is(2))==1)
 X(Is(2))=0;
 else
 X(Is(2))=1;
 end
 accum = 0;
 po2 = 1;
 for jw =BN:-1:1;
  if X(jw)==1;
  accum = accum +po2;
  Po2=Po2*2;
  else;
  Po2 = Po2*2;
  end;
 end ;
 W(it,Is(1))=accum;
 else
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
% NEW GENERATION ARRANGMENT
 for iq=3:ps;
   for iu=1:cl;
   u(iq,iu)=W(iq-2,iu);
   end
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
% STOPPING CRITERIA CHECK

   E(jmain)=min-top(jmain);
    if top(jmain)<min;
    min=top(jmain);;
    else
    end

 end   %MAIN LOOP OF THE PROGRAM

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

8. Second hybrid method between Tabu and GA

```
% This program applies a the second hybrid model of Tabu Search with
% Genetic Algorithm to six bus system parameters of the program are as
% follows:
%
% 1. Population Size (ps or mn) and they are set to be 15
% 2. Number of right-of-way (cl) and it is set to be 9
% 3. Generation number (GN) and it is set to be 1200
% 4. Cross-Over rate (pcross)and it is set to be 0.82
% 5. Mutation rate (pmute) and it is set to be 0.071
% 6. Degits Number of The Binary Number (BN) and
%    it is set to be 2
% 7. Size OF tabu list (tl) and it is set to be 9;
% 8. The Starting Global Minimum of The Objective
%    Function (min) it is set to 999999
%
% These parameters are set for six-bus system. If the
%  users want to change the system, they should change the
%  load-flow program lfax in this program to ld2x (for
%  IEEE-25 system) or lfsx for SCECO-East system. Also the
%  parameters specifed above should be changed to fit each system as
%  shown in the thesis.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Initial The Parameter of The Program
%Population Size
ps=15;
% number of right-of-way
cl=9;
% Cross-Over rate
pcross=.82 ;
% Mutation rate
pmute=.072;
% Degits Number of The Binary Number
BN=2;
% MOVEMENT NUMBER
mn=15;
% Generation number
GN1=1200;
% The Global Minimum of The Objective Function
min=99999;
% SIZE OF TABU LIST
tl=9;
%%%%%%%%%%%%%%%%%%%%
% INTIAL VALUES OF TABU LIST
xtl=rand(tl,cl+1);
%Initial population
u=4*rand(mn,cl);
for i=1:mn
```

```
  for j=1:cl
    for ii=0:10;
    if u(i,j)>ii&u(i,j)<ii+1;
    u(i,j)=ii;
    break
    else
    end
    end
  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION
 for jmain=1:GN1;
    for p=1:mn;
      for ia=1:cl;
      ROW(ia)=u(p,ia);
      end;
   lfax
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS

 for e1=1:mn
 for  k=1:mn-1;
  for  L=k+1,mn;
    if (OBJECT(k)>OBJECT(L));
    TEMP = OBJECT(k);
    OBJECT(k) = OBJECT(L);
    OBJECT(L) = TEMP;
    for jm =1:cl;
    TEM = u(k, jm);
    u(k,jm) = u(L,jm);
    u(L,jm) = TEM;
    end;
    else;
    end;
  end;
 end;
 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%
 % SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
 top(jmain)=OBJECT(1);
 for ie=1:cl;
 BROW(jmain,ie)=u(1,ie);
 end
 BROW(jmain,cl+1)=OBJECT(1);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%
 %CHECKING THE ACCEPTENCY TO THE TABU LIST
```

```
GR=0;S=0;
 for i=1:tl
  index=0;
   for j=1:cl
    if xtl(i,j)==BROW(jmain,j);
    index=index+1;
    else
    end
   end
    if index==(cl)
    S=S+1;
    else
    GR=GR+1;
    end
 end

  if GR==tl
     for i=tl:-1:2
        for j=1:cl+1
        xtl(i,j)=xtl(i-1,j);
        end
      end
     for j=1:cl+1
     xtl(1,j)=BROW(jmain,j);
     end
   else
   end
  clear GR S
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% NEW GENERATION PRODUCTION BY FOLLOWING
% 1.CROSSOVER SELECTION USING ROLETTE WHEEL SELECTION
 for j=1:ps;
  fit(j)=1/(1+OBJECT(j));
 end ;
 sumfit=sum(fit);
 %%%%%%%%%%%%%%%%%%%%%%
 for Ge=1:2:(ps-2)   % main loop of crossover implementation
  for m=1:2;
  partsum=0;
  Ran=rand*sumfit;
    for j=1:ps;
    partsum= partsum+fit(j);
    if (partsum >= Ran| j==ps);
    s(m)=j;
    break
    else
    end
    end
  end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  %%%%%%
```

```
% CROSSOVER IMPLEMENTATION

  if(pcross>rand)
  as(1)=cl* rand;
  as(2)=cl* rand;
         for ss=1:2;
           for ii=0:100;
       if as(ss)>ii & as(ss)<ii+1;
       as(ss)=ii+1;
       break
       else
       end
      end
     end
   else
  as(1)=1;
  as(2)=cl;
  end
 sort(as);

 for ic =1:cl;
  Parent1 (ic) = u(s(1),ic);
  Parent2 (ic) = u(s(2),ic);
 end

 for jc = 1:as(1)
child1(jc)=Parent1(jc);
child2(jc)=Parent2(jc);
 end

 %for jd = as(1): as(2)
 % child1(jd) =Parent2(jd);
 % child2(jd) =Parent1(jd);
 %end

  for  jc = as(1)+1:cl;
 child1(jc) = Parent1(jc);
 child2(jc) = Parent2(jc);
  end

  for ig=1:cl
 W(Ge,ig)=child1(ig);
 W(Ge+1,ig)=child2(ig);
  end

 clear child1 child2 Parent2 Parent1

 end

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%
 % MUTATION IMPLEMENTATION
```

```
for it=1:ps-2;
 if(pmute>rand);
 Is(1)= cl*rand;
 Is(2)= BN*rand;
   for ss= 1:2;
    for ii=0:100;
          if Is(ss)>ii & Is(ss)<ii+1;
          Is(ss)=ii+1;
          break;
          else;
          end;
      end;
    end;

IM=W(it,Is(1));
  for     jr=BN:-1:1;
     xx=IM/2;
    for i=0:100;
     if ((xx>i & xx<i+1) | xx == i);
     IM=i;
     R=xx-IM;
     break;
     else;
     end;
    end;
   if R==0;
   X(jr)=0;
   else;
   X(jr)=1;
   end;
  end;
 clear xx


  if (X(Is(2))==1)
  X(Is(2))=0;
  else
  X(Is(2))=1;
  end
  accum = 0;
  po2 = 1;
  for jw =BN:-1:1;
   if X(jw)==1;
   accum = accum +po2;
   Po2=Po2*2;
   else;
   Po2 = Po2*2;
   end;
  end ;
  W(it,Is(1))=accum;
  else
```

```
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NEW GENERATION ARRANGMENT
 for iq=3:ps;
  for iu=1:cl;
  u(iq,iu)=W(iq-2,iu);
  end
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STOPPING CRITEREIA CHECK
  E(jmain)=min-top(jmain);
   if top(jmain)<min;
   min=top(jmain);;
   for sssx=1:cl
   OPTIMUM(sssx)=BROW(jmain,sssx);
   end
   else
   end

  % if abs(E(jmain))< .01;
  % break
  % else
  % end
 end   %MAIN LOOP OF THE PROGRAM

min=min+30;
OPTIMUM(6)=OPTIMUM(6)+1;
```

9. A Hybrid method between Neural and Genetic

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program applies a hybrid of Neural Network and Genetic
% Algorithm to six bus system, parameters of the Program are as follows:
%
% A. Neural Network
% 1. State Space Size, training set, (ps) and it is set to be 50
% 2. Right-of-way (cl) and it is set to be 9
% 3. Total Iteration number (GN1)and it is set to be 1000
% 4. number of the generation in each iteration (pl1) and it is set to be 10
% 5. Input neuron is one, hidden nuerons are three and output neurons are
%    same as the size of (cl) which in this case is 9
% 6. The Starting Global Minimum of The Objective
%     Function (min) it is set to 999999
% B. Genetic Algorithm
% 1. Population Size (ps) and it is set to be 15
% 2. Number of right-of-way (cl) and it is set to be 9
% 3. Generation number (GN) and it is set to be 1200
% 4. Cross-Over rate (pcross)and it is set to be 0.82
% 5. Mutation rate (pmute) and it is set to be 0.071
% 6. Degits Number of The Binary Number (BN) and
%    it is set to be 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% These parameters are set for six-bus system, If the
% users want to change the system, they should change the
% load-flow program lfax in this program to ld2x (for
% IEEE-25 system) or lfsx for SCECO-East system. Also the
% parameters specifed above should be changed to fit each system as
% shown in the thesis.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% GLOBAL MINIMUM VALUES
K1=.0251;K2=1;
min=1000;
% STATE SPACE
ps=50;
% RIGHT OF WAY
cl=9;
%ITERATION
GN1=1000;
pl1=10;
pss=50;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%INITIAL SOLUTION STATES
u=4*rand(pss,cl);
for i=1:pss
   for j=1:cl
      for ii=0:10;
      if u(i,j)>ii&u(i,j)<ii+1;
```

```
        u(i,j)=ii;
        break
        else
        end
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION FOR
% THE INTIAL SOLUTION STATES
        for p=1:pss;
        for ia=1:cl;
        ROW(ia)=u(p,ia);
        end;
     lfax
     end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
for www=1:pss
for k=1:pss-1;
  for L=k+1,pss;
     if (OBJECT(k)>OBJECT(L));
     TEMP = OBJECT(k);
     OBJECT(k) = OBJECT(L);
     OBJECT(L) = TEMP;
     for jm =1:cl;
     TEM = u(k, jm);
     u(k,jm) = u(L,jm);
     u(L,jm) = TEM;
     end;
     else;
     end;
  end;
 end;
 end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
top(1)=OBJECT(1);
for ie=1:cl;
BROW(1,ie)=u(1,ie);
end
BROW(1,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STORING THE BEST SOLUTION
    if top(1)<min;
    min=top(1);
    else
    end

% APPLYING NEURAL NETWORK APPROACH TO HAVE A NEW SOLUTION STATES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART ONE GIVING THE TRAINING STATES
```

```
for qqq1=1:ps
 for qqq2=1:cl
T(qqq2,qqq1)=K1*u(qqq1,qqq2);
end
end
for qqq1=1:ps
st(qqq1)=K2*OBJECT(qqq1);
end
clear OBJECT u
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 for jmain=2:GN1
gt1=[T' st'];

tp=[100 1000 .6 .03];
% loop for getting the new soluion states
for rp=1:pl1
[w1,b1,w2,b2]=initff(st,4,'logsig',T,'purelin');
[w1,b1,w2,b2,epoch,tr]=trainbpx(w1,b1,'logsig',w2,b2,'purelin',st,T,tp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

d9=K2*.81*min*rand;
s=(1.8/K1)*simuff(d9,w1,b1,'logsig',w2,b2,'purelin');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for imain=1:1;
TQ1=cl*rand;
 for ii=0:100;
  if TQ1>ii & TQ1<ii+1
  TQ1=ii+1;
  break
  else
  end
  end
end
sr=abs(s'/1);

   for j=1:cl
       for ii= 0:20;
       if sr(j)>ii&sr(j)<ii+1;
       sr(j)=ii;
       break
       else
       end
       end
     end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 for j=1:cl
 ns(rp,j)=sr(j);
 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION FOR
```

```
% THE INTIAL SOLUTION STATES
     for p=1:pl1;
     for ia=1:cl;
     ROW(ia)=ns(p,ia);
     end;
   lfax
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
for www=1:pl1
for k=1:pl1-1;
 for L=k+1,pl1;
   if (OBJECT(k)>OBJECT(L));
   TEMP = OBJECT(k);
   OBJECT(k) = OBJECT(L);
   OBJECT(L) = TEMP;
   for jm =1:cl;
   TEM = ns(k, jm);
   ns(k,jm) = ns(L,jm);
   ns(L,jm) = TEM;
   end;
   else;
   end;
  end;
 end;
 end;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
 top(jmain)=OBJECT(1);
 for ie=1:cl;
 BROW(jmain,ie)=ns(1,ie);
 end
 BROW(jmain,cl+1)=OBJECT(1);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 ns=ns;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % STORING THE BEST SOLUTION
    if top(jmain)<min;
    min=top(jmain);
    for mx=1:cl;
    OPT(mx)=BROW(jmain,mx);
    end
    else
    end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  gt2=[ns OBJECT'];
  gt=[gt1;gt2];
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
  for www=1:(pl1+ps)
  for k=1:(pl1+ps)-1;
   for L=k+1,(pl1+ps);
```

```
    if (gt(k,cl+1)>gt(L,cl+1));
    TEMP = gt(k,cl+1);
    gt(k,cl+1) = gt(L,cl+1);
    gt(L,cl+1) = TEMP;
    for jm =1:cl+1;
    TEM = gt(k, jm);
    gt(k,jm) = gt(L,jm);
    gt(L,jm) = TEM;
    end;
    else;
    end;
  end;
end;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:ps
   for jj=1:cl
    u(ii,jj)=gt(ii,jj);
   end
end
T=K1*u';
%%%%%%%%%%%%
for ii=1:ps
st(ii)=K2*gt(ii,cl+1);
end
%%%%%%%%%%%
%main loop
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE BEST MOVE FROM THE NEURAL NETWORK
for ww=1:GN1
   Bm(ww)=BROW(ww,cl+1);
end
for op=1:GN1
for  k=1:GN1-1;
  for  L=k+1,GN1;
    if (Bm(k)>=Bm(L));
    TEMP = Bm(k);
    Bm(k) = Bm(L);
    Bm(L) = TEMP;
    for jm =1:cl+1;
    TEM = BROW(k, jm);
    BROW(k,jm) = BROW(L,jm);
    BROW(L,jm) = TEM;
    end;
    else;
    end;
  end;
 end;
 end;
 %%%%%%%%%%
```

```
% GENETIC ALGORITHM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initial The Parameter of The Program
%Population Size
ps=15;
% number of right-of-way
cl=9;
% Generation number
GN2=1200;
% Cross-Over rate
pcross=.82 ;
% Mutation rate
pmute=.075;
% Degits Number of The Binary Number
BN=2;
%New population for the GA
for ww=1:ps
  for qq=1:cl
   u(ww,qq)=BROW(ww,qq);
  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION
 for jmain=1:GN2;
    for p=1:ps;
      for ia=1:cl;
      ROW(ia)=u(p,ia);
      end;
    lfax
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
for  k=1:ps-1;
  for  L=k+1,ps;
    if (OBJECT(k)>OBJECT(L));
    TEMP = OBJECT(k);
    OBJECT(k) = OBJECT(L);
    OBJECT(L) = TEMP;
    for jm =1:cl;
    TEM = u(k, jm);
    u(k,jm) = u(L,jm);
    u(L,jm) = TEM;
    end;
    else;
    end;
  end;
 end;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
 top(jmain)=OBJECT(1);
 for ie=1:cl;
 BROWg(jmain,ie)=u(1,ie);
```

```
end
BROWg(jmain,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NEW GENERATION PRODUCTION BY FOLLOWING
% 1.CROSSOVER SELECTION USING ROLETTE WHEEL SELECTION
 for j=1:ps;
  fit(j)=1/(1+OBJECT(j));
 end  ;
 sumfit=sum(fit);
 %%%%%%%%%%%%%%%%%%%%%%%
 for Ge=1:2:(ps-2)   % main loop of crossover implementation
  for m=1:2;
  partsum=0;
  Ran=rand*sumfit;
    for j=1:ps;
    partsum= partsum+fit(j);
    if (partsum >= Ran| j==ps);
    s(m)=j;
    break
    else
    end
    end
 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 % CROSSOVER IMPLEMENTATION

     if(pcross>rand)
     as(1)=cl* rand;
     as(2)=cl* rand;
             for ss=1:2;
                for ii=0:100;
            if as(ss)>ii & as(ss)<ii+1;
            as(ss)=ii+1;
            break
            else
            end
          end
        end
     else
     as(1)=1;
     as(2)=cl;
     end
   sort(as);

    for ic =1:cl;
     Parent1 (ic) = u(s(1),ic);
     Parent2 (ic) = u(s(2),ic);
    end

    for jc = 1:as(1)
   child1(jc)=Parent1(jc);
   child2(jc)=Parent2(jc);
```

```
end

for jd = as(1): as(2)
 child1(jd) =Parent2(jd);
 child2(jd) =Parent1(jd);
end

 for jc = as(2):cl;
child1(jc) = Parent1(jc);
child2(jc) = Parent2(jc);
 end

 for ig=1:cl
W(Ge,ig)=child1(ig);
W(Ge+1,ig)=child2(ig);
 end

clear child1 child2 Parent2 Parent1

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MUTATION IMPLEMENTATION

 for it=1:ps-2;
  if(pmute>rand);
  Is(1)= cl*rand;
  Is(2)= BN*rand;
    for ss= 1:2;
     for ii=0:100;
          if Is(ss)>ii & Is(ss)<ii+1;
          Is(ss)=ii+1;
          break;
          else;
          end;
       end;
     end;

 IM=W(it,Is(1));
   for     jr=BN:-1:1;
     xx=IM/2;
     for i=0:100;
      if ((xx>i & xx<i+1) | xx == i);
      IM=i;
      R=xx-IM;
      break;
      else;
      end;
     end;
    if R==0;
    X(jr)=0;
    else;
```

```
  X(jr)=1;
  end;
 end;
clear xx


 if (X(Is(2))==1)
 X(Is(2))=0;
 else
 X(Is(2))=1;
 end
 accum = 0;
 po2 = 1;
 for jw =BN:-1:1;
  if X(jw)==1;
  accum = accum +po2;
  Po2=Po2*2;
  else;
  Po2 = Po2*2;
  end;
 end ;
 W(it,Is(1))=accum;
 else
 end
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NEW GENERATION ARRANGMENT
  for iq=3:ps;
   for iu=1:cl;
   u(iq,iu)=W(iq-2,iu);
   end
  end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STOPPING CRITERIA CHECK

   E(jmain)=min-top(jmain);
    if top(jmain)<min;
    min=top(jmain);;
    else
    end
   end   %MAIN LOOP OF THE PROGRAM
```

10. A Hybrid method between Neural, Tabu and Genetic

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% This program applies a hybrid of Neural Network with a second hybrid
% model between Tabu and Genetic to six bus system
%  Parameters of the Program are as follows:
% A. Neural Network
% 1. State Space Size, training set, (ps) and it is set to be 50
% 2. Right-of-way (cl) and it is set to be 9
% 3. Total Iteration number (GN1)and it is set to be 1000
% 4. number of the generation in each iteration (pl1) and it is set to be 10
% 5. Input neuron is one, hidden nuerons are three and output neurons are
%    same as the size of (cl) which in this case is 9
% 6. The Starting Global Minimum of The Objective
%    Function (min) it is set to 999999
%
% B. second hybrid mode between Tabu and Genetic
%  1. Population Size (ps or mn) and they are set to be 15
%  2. Generation number (GN2) and it is set to be 1200
%  3. Cross-Over rate (pcross)and it is set to be 0.82
%  4. Mutation rate (pmute) and it is set to be 0.071
%  5. Degits Number of The Binary Number (BN) and
%     it is set to be 2
%  6. Size OF tabu list (tl) and it is set to be 9;
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%


% GLOBAL MINIMUM VALUES
K1=.0251;K2=1;
min=1000;
% STATE SPACE
ps=50;
% RIGHT OF WAY
cl=9;
%ITERATION
GN1=1000;
pl1=10;
pss=50;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%INITIAL SOLUTION STATES
u=4*rand(pss,cl);
for i=1:pss
   for j=1:cl
      for ii=0:10;
      if u(i,j)>ii&u(i,j)<ii+1;
      u(i,j)=ii;
      break
```

```
        else
        end
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION FOR
% THE INTIAL SOLUTION STATES
        for p=1:pss;
        for ia=1:cl;
        ROW(ia)=u(p,ia);
        end;
    lfax
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
for  www=1:pss
for  k=1:pss-1;
  for  L=k+1,pss;
    if (OBJECT(k)>OBJECT(L));
    TEMP = OBJECT(k);
    OBJECT(k) = OBJECT(L);
    OBJECT(L) = TEMP;
    for jm =1:cl;
    TEM = u(k, jm);
    u(k,jm) = u(L,jm);
    u(L,jm) = TEM;
    end;
    else;
    end;
  end;
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
top(1)=OBJECT(1);
for ie=1:cl;
BROW(1,ie)=u(1,ie);
end
BROW(1,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STORING THE BEST SOLUTION
    if top(1)<min;
    min=top(1);
    else
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% APPLYING NEURAL NETWORK APPROACH TO HAVE A NEW SOLUTION STATES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART ONE GIVING THE TRAINING STATES
for qqq1=1:ps
  for qqq2=1:cl
```

```
T(qqq2,qqq1)=K1*u(qqq1,qqq2);
end
end
for qqq1=1:ps
st(qqq1)=K2*OBJECT(qqq1);
end
clear OBJECT u
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 for jmain=2:GN1
gt1=[T' st'];

tp=[100 1000 .6 .03];
% loop for getting the new soluion states
for rp=1:pl1
[w1,b1,w2,b2]=initff(st,4,'logsig',T,'purelin');
[w1,b1,w2,b2,epoch,tr]=trainbpx(w1,b1,'logsig',w2,b2,'purelin',st,T,tp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 d9=K2*.81*min*rand;
 s=(1.8/K1)*simuff(d9,w1,b1,'logsig',w2,b2,'purelin');
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 for imain=1:1;
 TQ1=cl*rand;
  for ii=0:100;
   if TQ1>ii & TQ1<ii+1
   TQ1=ii+1;
   break
   else
   end
   end
 end
 sr=abs(s'/1);

    for j=1:cl
        for ii= 0:20;
        if sr(j)>ii&sr(j)<ii+1;
        sr(j)=ii;
        break
        else
        end
        end
     end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  for j=1:cl
  ns(rp,j)=sr(j);
  end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  end
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  % STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION FOR
  % THE INTIAL SOLUTION STATES
        for p=1:pl1;
```

```
      for ia=1:cl;
      ROW(ia)=ns(p,ia);
      end;
   lfax
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
for www=1:pl1
for k=1:pl1-1;
 for L=k+1,pl1;
   if (OBJECT(k)>OBJECT(L));
   TEMP = OBJECT(k);
   OBJECT(k) = OBJECT(L);
   OBJECT(L) = TEMP;
   for jm =1:cl;
   TEM = ns(k, jm);
   ns(k,jm) = ns(L,jm);
   ns(L,jm) = TEM;
   end;
   else;
   end;
  end;
 end;
 end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
top(jmain)=OBJECT(1);
for ie=1:cl;
BROW(jmain,ie)=ns(1,ie);
end
BROW(jmain,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ns=ns;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STORING THE BEST SOLUTION
   if top(jmain)<min;
   min=top(jmain);
   for mx=1:cl;
   OPT(mx)=BROW(jmain,mx);
   end
   else
   end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 gt2=[ns OBJECT'];
 gt=[gt1;gt2];
 %%%%%%%%%%%%%%%%%%
 % SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS
 for www=1:(pl1+ps)
 for k=1:(pl1+ps)-1;
  for L=k+1,(pl1+ps);
    if (gt(k,cl+1)>gt(L,cl+1));
    TEMP = gt(k,cl+1);
```

```
gt(k,cl+1) = gt(L,cl+1);
gt(L,cl+1) = TEMP;
for jm =1:cl+1;
TEM = gt(k, jm);
gt(k,jm) = gt(L,jm);
gt(L,jm) = TEM;
end;
else;
end;
end;
end;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ii=1:ps
for jj=1:cl
u(ii,jj)=gt(ii,jj);
end
end
T=K1*u';
%%%%%%%%%%%
for ii=1:ps
st(ii)=K2*gt(ii,cl+1);
end
%%%%%%%%%%
%main loop
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SORTING THE BEST MOVE FROM THE NEURAL NETWORK
for ww=1:GN1
Bm(ww)=BROW(ww,cl+1);
end
for op=1:GN1
for k=1:GN1-1;
for L=k+1,GN1;
if (Bm(k)>=Bm(L));
TEMP = Bm(k);
Bm(k) = Bm(L);
Bm(L) = TEMP;
for jm =1:cl+1;
TEM = BROW(k, jm);
BROW(k,jm) = BROW(L,jm);
BROW(L,jm) = TEM;
end;
else;
end;
end;
end;
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SECOND HYBRIDATION METHOD BETWEEN TABU AND GENATIC
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initial The Parameter of The Program
%Population Size
ps=15;
% number of right-of-way
cl=9;
% Generation number
GN2=1200;
% Cross-Over rate
pcross=.82 ;
% Mutation rate
pmute=.072;
% Degits Number of The Binary Number
BN=2;
% MOVEMENT NUMBER
mn=15;
% SIZE OF TABU LIST
tl=9;
% INTIAL VALUES OF TABU LIST
xtl=rand(tl,cl+1);
%Initial population
for i=1:mn
   for j=1:cl
      u(i,j)=BROW(i,j);
   end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% STATRING THE COMPUTATION OF THE OBJECTIVE FUNCTION
 for jmain=1:GN2;
     for p=1:mn;
       for ia=1:cl;
       ROW(ia)=u(p,ia);
       end;
     lfax
     end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% SORTING THE ARRAY OF THE OBJECTIVE FUNCTION WITH ITS POSSIBLE ROUTS

 for e1=1:mn
 for  k=1:mn-1;
  for  L=k+1,mn;
    if (OBJECT(k)>OBJECT(L));
    TEMP = OBJECT(k);
    OBJECT(k) = OBJECT(L);
    OBJECT(L) = TEMP;
    for jm =1:cl;
    TEM = u(k, jm);
    u(k,jm) = u(L,jm);
    u(L,jm) = TEM;
    end;
```

```
   else;
   end;
  end;
end;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% SAVING THE BEST OBJECTIVE FUNCTION VALUE AND ITS ROUT
top(jmain)=OBJECT(1);
for ie=1:cl;
BROWh(jmain,ie)=u(1,ie);
end
BROWh(jmain,cl+1)=OBJECT(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CHECKING THE ACCEPTENCY TO THE TABU LIST
GR=0;S=0;
 for i=1:tl
  index=0;
    for j=1:cl
     if xtl(i,j)==BROWh(jmain,j);
     index=index+1;
     else
     end
    end
     if index==(cl)
     S=S+1;
     else
     GR=GR+1;
     end
  end

  if GR==tl
     for i=tl:-1:2
        for j=1:cl+1
        xtl(i,j)=xtl(i-1,j);
        end
      end
     for j=1:cl+1
     xtl(1,j)=BROWh(jmain,j);
     end
   else
   end
 clear GR S
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%
 % NEW GENERATION PRODUCTION BY FOLLOWING
 % 1.CROSSOVER SELECTION USING ROLETTE WHEEL SELECTION
  for j=1:ps;
   fit(j)=1/(1+OBJECT(j));
  end  ;
 sumfit=sum(fit);
```

```
%%%%%%%%%%%%%%%%%%%
for Ge=1:2:(ps-2)   % main loop of crossover implementation
 for m=1:2;
 partsum=0;
 Ran=rand*sumfit;
  for j=1:ps;
  partsum= partsum+fit(j);
  if (partsum >= Ran| j==ps);
  s(m)=j;
  break
  else
  end
  end
 end
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%
 % CROSSOVER IMPLEMENTATION

    if(pcross>rand)
    as(1)=cl* rand;
    as(2)=cl* rand;
           for ss=1:2;
             for ii=0:100;
          if as(ss)>ii & as(ss)<ii+1;
          as(ss)=ii+1;
          break
          else
          end
         end
       end
     else
    as(1)=1;
    as(2)=cl;
    end
  sort(as);

   for ic =1:cl;
    Parent1 (ic) = u(s(1),ic);
    Parent2 (ic) = u(s(2),ic);
   end

   for jc = 1:as(1)
  child1(jc)=Parent1(jc);
  child2(jc)=Parent2(jc);
   end

    for jd = as(1): as(2)
     child1(jd) =Parent2(jd);
     child2(jd) =Parent1(jd);
    end

    for  jc = as(2):cl;
```

```
child1(jc) = Parent1(jc);
child2(jc) = Parent2(jc);
 end

 for ig=1:cl
W(Ge,ig)=child1(ig);
W(Ge+1,ig)=child2(ig);
 end

clear child1 child2 Parent2 Parent1

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% MUTATION IMPLEMENTATION

 for it=1:ps-2;
  if(pmute>rand);
  Is(1)= cl*rand;
  Is(2)= BN*rand;
    for ss= 1:2;
     for ii=0:100;
          if Is(ss)>ii & Is(ss)<ii+1;
          Is(ss)=ii+1;
          break;
          else;
          end;
      end;
     end;

 IM=W(it,Is(1));
   for      jr=BN:-1:1;
     xx=IM/2;
     for i=0:100;
      if ((xx>i & xx<i+1) | xx == i);
      IM=i;
      R=xx-IM;
      break;
      else;
      end;
     end;
    if R==0;
    X(jr)=0;
    else;
    X(jr)=1;
    end;
   end;
  clear xx


   if (X(Is(2))==1)
```

```
X(Is(2))=0;
else
X(Is(2))=1;
end
accum = 0;
po2 = 1;
for jw =BN:-1:1;
 if X(jw)==1;
 accum = accum +po2;
 Po2=Po2*2;
 else;
 Po2 = Po2*2;
 end;
 end ;
 W(it,Is(1))=accum;
 else
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NEW GENERATION ARRANGMENT
 for iq=3:ps;
  for iu=1:cl;
  u(iq,iu)=W(iq-2,iu);
  end
 end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% STOPPING CRITEREIA CHECK
  E(jmain)=min-top(jmain);
   if top(jmain)<min;
   min=top(jmain);;
   for sss=1:cl
   OPTIMUM(sss)=BROWh(jmain,sss);
   end
   else
   end

 end   %MAIN LOOP OF THE PROGRAM
```

11. Linear and Quadratic Programming model for 6-bus system

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
% This program applies a Linear programming and quadratic programming
% to six bus system, the parameters of the formulation are as :
% A. Quadratic Programming:
%   1. The Impedance of the system is specifed in matrix h
%   2. The KCL and KVL of the system is specified in matrix a and b
%   3. The prices of the new added lines are specified in f
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sample of the solution run
% ***** [x lambda]=qp(1000*h,1.25*f,a,b,[],[],zeros(13,1))
%
%
% B. Linear Programming:
%  all the above parameters are used except the Impedance of the system
%  (h) is set to be zero.
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ****** [x lambda]=qp(0*h,1.25*f,a,b,-2*ones(13,1),2*ones(13,1),zeros(13,1))
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%

h=[.05 0 0 0 0 0 0 0;
0 .05 0 0 0 0 0 0 ;
0 0 .1 0 0 0 0 0;
0 0 0 .05 0 0 0 0;
0 0 0 0 .15 0 0 0;
0 0 0 0 0 .10 0 0
0 0 0 0 0 0  .000001 0
0 0 0 0 0 0  0 .000001];
h(9,9)=.0000001;
h(10,10)=.000005
h(11,11)=.0000075
h(12,12)=.0000075
h(13,13)=.0000012
f=[0 0 0 0 0 0 0 0 0  20 30 30 61];
a=[1  0  1  0  1  0  0  0  0  0  0  0  0;
  0  0  -1  -1  0  -1  0  0  0  0  0  1  0;
  0  -1  0  1  0  0  0  0  0  -1  0  0  0;
  0  0  0  0  -1  1  0  0  0  0  1  0  0;
  -1  1  0  0  0  0  0  0  0  1  0  0  1;
  0  0  0  0  0  0  0  0  0  0  -1  -1  -1;
  .2  .2  -.4  .2  0  0  0  0  0  0  0  0  0;
  0  0  .4  0  -.6  -.4  0  0  0  0  0  0  0]

b=[.3;2.4;-1.25;1.6;2.4;-5.45;0;0]

[x lambda]=qp(1000*h,1.25*f,a,b,[],[],zeros(13,1))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%

h=[.05 0 0 0 0 0 0 0 0 0;
```

```
0 .05 0 0 0 0 0 0 0 0 0 ;
0 0 .1 0 0 0 0 0 0 0;
0 0 0 .05 0 0 0 0 0 0;
0 0 0 0 .15 0 0 0 0 0;
0 0 0 0 0 .10 0 0 0 0
0 0 0 0 0 0  .0750 0 0 0
0 0 0 0 0 0  0 .075*2 0 0];
h(9,9)=.000000012;
h(10,10)=0.0000075;
h(11,11)=0.0000075;
h(12,12)=.00000075;
h(13,13)=.00000012;

f=[0 0 0 0 0 0 0 0 0 20 30 30 61];
a=[1   0   1   0   1   0   0   0    0    0   0   0   0;
   0   0  -1  -1   0  -1   0   2    0    0   0   1   0;
   0  -1   0   1   0   0   0   0    0   -1   0   0   0;
   0   0   0   0  -1   1   1   0    0    0   1   0   0;
  -1   1   0   0   0   0   0   0    0    1   0   0   1;
   0   0   0   0   0   0  -1  -2    0    0  -1  -1  -1;
  .2  .2 -.4  .2   0   0   0   0    0    0   0   0   0;
   0   0  .4   0 -.6 -.4   0   0    0    0   0   0   0;
   0   0   0   0   0  .4 -.3  .6    0    0   0   0   0];

b=[.3;2.4;-1.25;1.6;2.4;-5.45;0;0;0]

[x lambda]=qp(1000*h,2.25*f,a,b,[],[],zeros(13,1))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h=[.05 0 0 0 0 0 0 0 0 0;
0 .05*2 0 0 0 0 0 0 0 0 ;
0 0 .1 0 0 0 0 0 0 0;
0 0 0 .05 0 0 0 0 0 0;
0 0 0 0 .15 0 0 0 0 0;
0 0 0 0 0 .10 0 0 0 0
0 0 0 0 0 0  .075*2 0 0 0
0 0 0 0 0 0  0 .075*3 0 0];
h(9,9)=.12;
h(10,10)=0.0000075;
h(11,11)=0.0000075;
h(12,12)=.00000075;
h(13,13)=.00000012;

f=[0 0 0 0 0 0 0 0 0 20 30 30 61];
a=[1   0   1   0   1   0   0   0    0    0   0   0   0;
   0   0  -1  -1   0  -1   0   3    0    0   0   1   0;
   0  -2   0   1   0   0   0   0    0   -1   0   0   0;
   0   0   0   0  -1   1   2   0    0    0   1   0   0;
  -1   1   0   0   0   0   0`  0    1    1   0   0   1;
   0   0   0   0   0   0  -2  -3   -1    0  -1  -1  -1;
  .2  .4 -.4  .2   0   0   0   0    0    0   0   0   0;
   0   0  .4   0 -.6 -.4   0   0    0    0   0   0   0;
```

```
 0  0  0  0  0 .4  -.6 .9   0   0   0   0   0];

b=[.3;2.4;-1.25;1.6;2.4;-5.45;0;0;0]

[x lambda]=qp(1000*h,2.25*f,a,b,[],[],zeros(13,1))

%%%%%%%%%%%%%%%%%%%%%%%%%%

h=[.05 0 0 0 0 0 0 0 0 0 0;
0 .05*2 0 0 0 0 0 0 0 0 ;
0 0 .1 0 0 0 0 0 0 0;
0 0 0 .05 0 0 0 0 0 0;
0 0 0 0 .15 0 0 0 0 0;
0 0 0 0 0 .10 0 0 0 0
0 0 0 0 0 0  .075*3 0 0 0
0 0 0 0 0 0  0 .075*4 0 0];
h(9,9)=.12*1;
h(10,10)=0.00005;
h(11,11)=0.000075;
h(12,12)=.0000075;
h(13,13)=.0000012;

f=[0 0 0 0 0 0 0 0 0 20 30 30 61];
a=[1   0   1   0   1   0   0   0   0   0   0   0   0;
  0   0  -1  -1   0  -1   0   4   0   0   0   1   0;
  0  -2   0   1   0   0   0   0   0  -1   0   0   0;
  0   0   0   0  -1   1   3   0   0   0   1   0   0;
 -1   1   0   0   0   0   0   0   1   1   0   0   1;
  0   0   0   0   0   0  -3  -4  -1   0  -1  -1  -1;
 .2  .4  -.4  .2   0   0   0   0   0   0   0   0   0;
  0   0  .4   0  -.6 -.4   0   0   0   0   0   0   0;
  0  -.4   0  -.2   0   0   0  -1.2 .48  0   0   0   0;
  0   0   0   0   0  .4  -.9  1.2  0   0   0   0   0];
b=[.3;2.4;-1.25;1.6;2.4;-5.45;0;0;0;0]

[x lambda]=qp(1000*h,2.25*f,a,b,[],[],zeros(13,1))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

12. Linear and Quadratic Program for 25-Bus System

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% This program applies a Linear programming and quadratic programming
% to IEEE-25 bus system, the parameters of the formulation are as :
% A. Quadratic Programming:
%   1. The Impedance of the system is specifed in matrix h
%   2. The KCL and KVL of the system is specified in matrix a and b
%   3. The prices of the new added lines are specified in f
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sample of the solution run
% **** [x lambda]=qp(1e5*H,(.2)*f,a,1.65*b,[],[],zeros(72,1),25)
%
%
% B. Linear Programming:
%   all the above parameters are used except the Impedance of the system
%   (h) is set to be zero.
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% **** [x lambda]=qp(1e5*H,(.2)*f,a,1.65*b,-4*ones(72,1),4*ones(72,1),zeros(72,1),25)
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:36
t(i)=0;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A12=[-1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
   0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 1 0 0 0 0 0 0 0 0 0 0 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0
   0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 -1 -1 0 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 -1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 1 0 0 0 0 0
   0 0 0 0 0 -1 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 -1 0 0 0 0
   0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -1 0 0 0
   0 0 0 0 0 0 0 0 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 -1 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
   0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 -1 1 0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 -1 1 0 0 0 0
```

```
  0  0  0 0 0 0 0 0 0 0 0 -1  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
  0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1] ;
%%%%%%%%%%%%%%%%%%%%%%%%%
A11=[-(1+t(1))  -(1+t(2))  -(1+t(3))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 0 0 0
   0 (1+t(1))   0 -(1+t(4))  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
   0  0  0  (1+t(4))  -(1+t(5)) 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0 0 0  (1+t(6))  (1+t(7))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0 0 0 0 0  (1+t(8)) (1+t(9))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
(1+t(35))  0
   0  0  0 0 0 0 0 0 0  (1+t(10)) (1+t(11)) (1+t(12))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 0 0 0
   0  (1+t(2))  0  0  0  0  0  0  0  0  0  0  -(1+t(13)) -(1+t(14)) 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 0 0 0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 -(1+t(15)) (1+t(16))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  (1+t(17))  -(1+t(18)) 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -(1+t(19)) -(1+t(20))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0
   0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 -(1+t(17))  0  (1+t(18))  0 -(1+t(21))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0  (1+t(22)) -(1+t(23))  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  (1+t(3)) 0  0  0  0  0  0  0  0  0  (1+t(13))  0  0  0  0  0  0  0  0  0  0 -(1+t(24)) -(1+t(25))  0  0  0  0
0 0 0 0 0 0 0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  (1+t(21)) -(1+t(22))  0  0  0  (1+t(26))  0  0  0  0  0  0
0 0 0 0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  (1+t(18))  0  (1+t(20))  0  0  0  0  0  0 -(1+t(27))  0  0  0  0  0
0 0 0 0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0  (1+t(14))  (1+t(15))  0  0  0  0  0  0  0  0  0  0  0  0 -(1+t(28)) -
(1+t(29))  0  0  0  0  0  0  0
   0  0  0 0 0 0 0 -(1+t(8))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -(1+t(30))  (1+t(31))  0
0 0 0 0
   0  0  0 0 0 -(1+t(6))  0  0  0 -(1+t(10))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  (1+t(24))  0  0  0  (1+t(28))  0
0  0 -(1+t(32))  0  0  0  0
   0  0  0 0 0 0 -(1+t(7))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  (1+t(30)) 0  0  -
(1+t(33))  0  0  0
0 0 0 0 0 0 0 0 -(1+t(9)) 0 -(1+t(11)) 0 0 0 0 0 0 0 0 0 0 0 0 0  (1+t(25)) 0 0 0 (1+t(29)) 0 0 0 0 -(1+t(34))  0
0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  (1+t(33)) (1+t(34))  0  0
   0 0 0 0  (1+t(5)) 0 0 0 0 0 0 0 0 0 0 -(1+t(16))  0  0  0  0 0 0 0 0  0 -(1+t(26))  (1+t(27))  0 0 0 0 0 0 0 0 0
   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 0  (1+t(23))  0  0  0  0  0  0  0 -(1+t(32))  (1+t(33))  0  0  0  0
   0  0  0 0 0 0 0 0 0 0 0 -(1+t(12))  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  (1+t(36))
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -(1+t(35)) -(1+t(36))] ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%
A21=3*[ .011 -.086 0 .02 .023 0 0  0  0  0  0  0  0  0 -.047 .041 .039 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0 0 0
   0 .086 -.096 0  0  0  0  0  0  0  0  0  0 .0476 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0 0 0 0 0 0 0  .165 -.165  0  0  0  0  0  0  0  0  0  0  0  0  0 .084 -.084  0  0  0  0  0  0  0  0  0  0  0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 .013 .014 .068 -.105  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
   0  0  0 0 0 -.104 .126   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 .014 .21 .119   0  0  0  0
   0  0  0 0 0 0 0 0 0 0 0 0 0 0 0 -.042 -.04   0  0  0  0  0  .051  .084  0  0  .017  0  -.084  0  0  0  -.119
0 0 0 0 ];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A22(6,36)=0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%a=[A11 A12; A21 A22];
%a=[A11 ; A21];
a=[A11 A12];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h(1,1)=.0027; h(2,2)=.0217; h(3,3)=.0236; h(4,4)=.005; h(5,5)=.0058;
h(6,6)=.026 ; h(7,7)=.0316; h(8,8)=.0212; h(9,9)=.0221; h(10,10)=.0417;
h(11,11)= .0417; h(12,12)=.0153; h(13,13)=.0119; h(14,14)=.0119;
h(15,15)=.0104; h(16,16)=.0097; h(17,17)=.0032; h(18,18)=.0036;
h(19,19)=.0169; h(20,20)=.0259; h(21,21)=.0061; h(22,22)=2*.013;
h(23,23)= .0210; h(24,24)=.021; h(25,25)=.021; h(26,26)= .0043;
h(27,27)=.0065; h(28,28)=.021; h(29,29)= .021; h(30,30)=.0035 ;
h(31,31)=.0532; h(32,32)=.0298 ; h(33,33)=.0481; h(34,34)=.0152;
h(35,35)=.0451;h(36,36)=.0226;
for i=1:36
h1(i,i)=0.001;
end
for i=1:36
h(i,i)=h(i,i)*(t(i)+1);
end
w(36,36)=0;
H=[h w; w h1];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
PRICE=[1410 1.*10428 1.25*11613 2666 1.15*3070 1840 2240 1495 1564 2916 2911 1104 1.245*6235
1.500*6303 5547 .15*5160 1720 1917 8216 12720 3190 6343 .15*253 1*253 1*253 .0001*2236 3466 253
253 186 .42*3330 .2*2102 .02*3397 842 1150 644 ];
sw(36)=0;
f=[.1*PRICE PRICE]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

b=[-5.3 1.28 1.81 0.74 0.71 0.71 -3.29 1.94 -0.67 -3.0 -4.0 0.0 0.0 2.74 0.0 0.0 0.68 1.75 0.57 0.0 1.36 -0.55
1.80 0.65 -3.30]
%%%
[x lambda]=qp(1e5*H,(.2)*f,a,1.65*b,[],[],zeros(72,1),25)
%%%%%
for i=1:36
y(i)=(t(i)+1)*x(i)+x(i+36);
end
%%%%%%
MAX=[8 .65 1 5 2 10 2.5 8 9.4 4.4 2.8 10.8 2.5 .9  4.9 .65 2.6 2.5 8 2.5 7 1 .7 1 2.5 2 3.6 2.5 5.64 4 3.5 1.5
1.1 1.8 2.2 2.2];
% over load
for i=1:36
O(i)=MAX(i)*(t(i)+1)-abs(y(i));
end;
O
```